MOBILE DEVICE

CROSS REFERENCE TO RELATED APPLICATION

[0001] The present application claims the benefits of and priority, under 35 U.S.C. § 119(e), to U.S. Provisional Application Serial Nos. 61/389,000, filed October 1, 2010, entitled "DUAL DISPLAY WINDOWING SYSTEM;" 61/389,117, filed October 1, 2010, entitled "MULTI-OPERATING SYSTEM PORTABLE DOCKETING DEVICE;" 61/389,087, filed October 1,2010, entitled "TABLET COMPUTING USER INTERFACE;"61/458,150, filed November 17, 2010, entitled "Dual Screen Email Client;" XX/XXXXXX, filed September XX, 2011, entitled "MOBILE DEVICE." Each of the aforementioned documents is incorporated herein by this reference in their entirety for all that they teach and for all purposes.

BACKGROUND

[0002] A substantial number of handheld computing devices, such as cellular phones, tablets, and E-Readers, make use of a touch screen display not only to deliver display information to the user but also to receive inputs from user interface commands. While touch screen displays may increase the configurability of the handheld device and provide a wide variety of user interface options, this flexibility typically comes at a price. The dual use of the touch screen to provide content and receive user commands, while flexible for the user, may obfuscate the display and cause visual clutter, thereby leading to user frustration and loss of productivity.

[0003] The small form factor of handheld computing devices requires a careful balancing between the displayed graphics and the area provided for receiving inputs. On the one hand, the small display constrains the display space, which may increase the difficulty of interpreting actions or results. On the other, a virtual keypad or other user interface scheme is superimposed on or positioned adjacent to an executing application, requiring the application to be squeezed into an even smaller portion of the display.

[0004] This balancing act is particularly difficult for single display touch screen devices. Single display touch screen devices are crippled by their limited screen space. When users are entering information into the device, through the single display, the ability to interpret information in the display can be severely hampered, particularly when a complex interaction between display and interface is required.

SUMMARY

[0005] There is a need for a dual multi-display handheld computing device that provides for enhanced power and/or versatility compared to conventional single display handheld computing devices. These and other needs are addressed by the various aspects, embodiments, and/or configurations of the present disclosure. Also, while the disclosure is presented in terms of exemplary embodiments, it should be appreciated that individual aspects of the disclosure can be separately claimed.

[0006] INSERT INDEPENDENT CLAIMS HERE

[0007] The present disclosure can provide a number of advantages depending on the particular aspect, embodiment, and/or configuration. ENTER ADVANTAGES HERE

[0008] These and other advantages will be apparent from the disclosure.

[0009] The phrases "at least one", "one or more", and "and/or" are open-ended expressions that are both conjunctive and disjunctive in operation. For example, each of the expressions "at least one of A, B and C", "at least one of A, B, or C", "one or more of A, B, and C", "one or more of A, B, or C" and "A, B, and/or C" means A alone, B alone, C alone, A and B together, A and C together, B and C together, or A, B and C together.

[0010] The term "a" or "an" entity refers to one or more of that entity. As such, the terms "a" (or "an"), "one or more" and "at least one" can be used interchangeably herein. It is also to be noted that the terms "comprising", "including", and "having" can be used interchangeably.

[0011] The term "automatic" and variations thereof, as used herein, refers to any process or operation done without material human input when the process or operation is performed. However, a process or operation can be automatic, even though performance of the process or operation uses material or immaterial human input, if the input is received before performance of the process or operation. Human input is deemed to be material if such input influences how the process or operation will be performed. Human input that consents to the performance of the process or operation is not deemed to be "material".

[0012] The term "computer-readable medium" as used herein refers to any tangible storage and/or transmission medium that participate in providing instructions to a processor for execution. Such a medium may take many forms, including but not limited to, non-volatile media, volatile media, and transmission media. Non-volatile media includes, for example, NVRAM, or magnetic or optical disks. Volatile media includes dynamic memory, such as main memory. Common forms of computer-readable media include, for example, a floppy disk, a

flexible disk, hard disk, magnetic tape, or any other magnetic medium, magneto-optical medium, a CD-ROM, any other optical medium, punch cards, paper tape, any other physical medium with patterns of holes, a RAM, a PROM, and EPROM, a FLASH-EPROM, a solid state medium like a memory card, any other memory chip or cartridge, a carrier wave as described hereinafter, or any other medium from which a computer can read. A digital file attachment to e-mail or other self-contained information archive or set of archives is considered a distribution medium equivalent to a tangible storage medium. When the computer-readable media is configured as a database, it is to be understood that the database may be any type of database, such as relational, hierarchical, object-oriented, and/or the like. Accordingly, the disclosure is considered to include a tangible storage medium or distribution medium and prior art-recognized equivalents and successor media, in which the software implementations of the present disclosure are stored.

[0013] The term "desktop" refers to a metaphor used to portray systems. A desktop is generally considered a "surface" that typically includes pictures, called icons, widgets, folders, etc. that can activate show applications, windows, cabinets, files, folders, documents, and other graphical items. The icons are generally selectable to initiate a task through user interface interaction to allow a user to execute applications or conduct other operations.

[0014] The term "screen," "touch screen," or "touchscreen" refers to a physical structure that includes one or more hardware components that provide the device with the ability to render a user interface and/or receive user input. A screen can encompass any combination of gesture capture region, a touch sensitive display, and/or a configurable area. The device can have one or more physical screens embedded in the hardware. However a screen may also include an external peripheral device that may be attached and detached from the device. In embodiments, multiple external devices may be attached to the device. Thus, in embodiments, the screen can enable the user to interact with the device by touching areas on the screen and provides information to a user through a display. The touch screen may sense user contact in a number of different ways, such as by a change in an electrical parameter (e.g., resistance or capacitance), acoustic wave variations, infrared radiation proximity detection, light variation detection, and the like. In a resistive touch screen, for example, normally separated conductive and resistive metallic layers in the screen pass an electrical current. When a user touches the screen, the two layers make contact in the contacted location, whereby a change in electrical field is noted and the coordinates of the contacted location calculated. In a capacitive touch screen, a capacitive

layer stores electrical charge, which is discharged to the user upon contact with the touch screen, causing a decrease in the charge of the capacitive layer. The decrease is measured, and the contacted location coordinates determined. In a surface acoustic wave touch screen, an acoustic wave is transmitted through the screen, and the acoustic wave is disturbed by user contact. A receiving transducer detects the user contact instance and determines the contacted location coordinates.

[0015] The term "display" refers to a portion of one or more screens used to display the output of a computer to a user. A display may be a single-screen display or a multi-screen display, referred to as a composite display. A composite display can encompass the touch sensitive display of one or more screens. A single physical screen can include multiple displays that are managed as separate logical displays. Thus, different content can be displayed on the separate displays although part of the same physical screen.

[0016] The term "displayed image" refers to an image produced on the display. A typical displayed image is a window or desktop. The displayed image may occupy all or a portion of the display.

[0017] The term "display orientation" refers to the way in which a rectangular display is oriented by a user for viewing. The two most common types of display orientation are portrait and landscape. In landscape mode, the display is oriented such that the width of the display is greater than the height of the display (such as a 4:3 ratio, which is 4 units wide and 3 units tall, or a 16:9 ratio, which is 16 units wide and 9 units tall). Stated differently, the longer dimension of the display is oriented substantially horizontal in landscape mode while the shorter dimension of the display is oriented substantially vertical. In the portrait mode, by contrast, the display is oriented such that the width of the display is less than the height of the display. Stated differently, the shorter dimension of the display is oriented substantially horizontal in the portrait mode while the longer dimension of the display is oriented substantially vertical.

[0018] The term "composited display" refers to a logical structure that defines a display that can encompass one or more screens. A multi-screen display can be associated with a composite display that encompasses all the screens. The composite display can have different display characteristics based on the various orientations of the device.

[0019] The term "gesture" refers to a user action that expresses an intended idea, action, meaning, result, and/or outcome. The user action can include manipulating a device (e.g.,

GOOGLE EXHIBIT 1015
Google and Motorola Mobility v. Multifold
IPR2025-00059

opening or closing a device, changing a device orientation, moving a trackball or wheel, etc.), movement of a body part in relation to the device, movement of an implement or tool in relation to the device, audio inputs, etc. A gesture may be made on a device (such as on the screen) or with the device to interact with the device.

[0020] The term "module" as used herein refers to any known or later developed hardware, software, firmware, artificial intelligence, fuzzy logic, or combination of hardware and software that is capable of performing the functionality associated with that element.

[0021] The term "gesture capture" refers to a sense or otherwise a detection of an instance and/or type of user gesture. The gesture capture can occur in one or more areas of the screen, A gesture region can be on the display, where it may be referred to as a touch sensitive display or off the display where it may be referred to as a gesture capture area.

[0022] A "multi-screen application" refers to an application that is capable of multiple modes. The multi-screen application mode can include, but is not limited to, a single screen mode (where the application is displayed on a single screen) or a composite display mode (where the application is displayed on two or more screens). A multi-screen application can have different layouts optimized for the mode. Thus, the multi-screen application can have different layouts for a single screen or for a composite display that can encompass two or more screens. The different layouts may have different screen/display dimensions and/or configurations on which the user interfaces of the multi-screen applications can be rendered. The different layouts allow the application to optimize the application's user interface for the type of display, *e.g.,* single screen or multiple screens. In single screen mode, the multi-screen application may present one window pane of information. In a composite display mode, the multi-screen application may present multiple window panes of information or may provide a larger and a richer presentation because there is more space for the display contents. The multi-screen applications may be designed to adapt dynamically to changes in the device and the mode depending on which display (single or composite) the system assigns to the multi-screen application. In alternative embodiments, the user can use a gesture to request the application transition to a different mode, and, if a display is available for the requested mode, the device can allow the application to move to that display and transition modes.

[0023] A "single-screen application" refers to an application that is capable of single screen mode. Thus, the single-screen application can produce only one window and may not be capable

of different modes or different display dimensions. A single-screen application is incapable of the several modes discussed with the multi-screen application.

[0024] The term "window" refers to a, typically rectangular, displayed image on at least part of a display that contains or provides content different from the rest of the screen. The window may obscure the desktop.

[0025] The terms "determine", "calculate" and "compute," and variations thereof, as used herein, are used interchangeably and include any type of methodology, process, mathematical operation or technique.

[0026] It shall be understood that the term "means" as used herein shall be given its broadest possible interpretation in accordance with 35 U.S.C., Section 112, Paragraph 6. Accordingly, a claim incorporating the term "means" shall cover all structures, materials, or acts set forth herein, and all of the equivalents thereof. Further, the structures, materials or acts and the equivalents thereof shall include all those described in the summary of the invention, brief description of the drawings, detailed description, abstract, and claims themselves.

[0027] The preceding is a simplified summary of the disclosure to provide an understanding of some aspects of the disclosure. This summary is neither an extensive nor exhaustive overview of the disclosure and its various aspects, embodiments, and/or configurations. It is intended neither to identify key or critical elements of the disclosure nor to delineate the scope of the disclosure but to present selected concepts of the disclosure in a simplified form as an introduction to the more detailed description presented below. As will be appreciated, other aspects, embodiments, and/or configurations of the disclosure are possible utilizing, alone or in combination, one or more of the features set forth above or described in detail below.

BRIEF DESCRIPTION OF THE DRAWINGS

[0028] FIG. 1A includes a first view of an embodiment of a multi-screen user device;

[0029] FIG. 1B includes a second view of an embodiment of a multi-screen user device;

[0030] FIG. 1C includes a third view of an embodiment of a multi-screen user device;

[0031] FIG. 1D includes a fourth view of an embodiment of a multi-screen user device;

[0032] FIG. 1E includes a fifth view of an embodiment of a multi-screen user device;

[0033] FIG. 1F includes a sixth view of an embodiment of a multi-screen user device;

[0034] FIG. 1G includes a seventh view of an embodiment of a multi-screen user device;

[0035] FIG. 1H includes a eighth view of an embodiment of a multi-screen user device;

[0036] FIG. 1I includes a ninth view of an embodiment of a multi-screen user device;

[0037] FIG. 1J includes a tenth view of an embodiment of a multi-screen user device;

[0038] FIG. 2 is a block diagram of an embodiment of the hardware of the device;

[0039] FIG. 3A is a block diagram of an embodiment of the state model for the device based on the device's orientation and/or configuration;

[0040] FIG. 3B is a table of an embodiment of the state model for the device based on the device's orientation and/or configuration;

[0041] FIG. 4A is a first representation of an embodiment of user gesture received at a device;

[0042] FIG. 4B is a second representation of an embodiment of user gesture received at a device;

[0043] FIG. 4C is a third representation of an embodiment of user gesture received at a device;

[0044] FIG. 4D is a fourth representation of an embodiment of user gesture received at a device;

[0045] FIG. 4E is a fifth representation of an embodiment of user gesture received at a device;

[0046] FIG. 4F is a sixth representation of an embodiment of user gesture received at a device;

[0047] FIG. 4G is a seventh representation of an embodiment of user gesture received at a device;

[0048] FIG. 4H is a eighth representation of an embodiment of user gesture received at a device;

[0049] FIG. 5A is a block diagram of an embodiment of the device software and/or firmware;

[0050] FIG. 5B is a second block diagram of an embodiment of the device software and/or firmware;

[0051] FIG. 6A is a first representation of an embodiment of a device configuration generated in response to the device state;

[0052] FIG. 6B is a second representation of an embodiment of a device configuration generated in response to the device state;

[0053] FIG. 6C is a third representation of an embodiment of a device configuration generated in response to the device state;

[0054] FIG. 6D is a fourth representation of an embodiment of a device configuration generated in response to the device state;

[0055] FIG. 6E is a fifth representation of an embodiment of a device configuration generated in response to the device state;

[0056] FIG. 6F is a sixth representation of an embodiment of a device configuration generated in response to the device state;

[0057] FIG. 6G is a seventh representation of an embodiment of a device configuration generated in response to the device state;

[0058] FIG. 6H is a eighth representation of an embodiment of a device configuration generated in response to the device state;

[0059] FIG. 6I is a ninth representation of an embodiment of a device configuration generated in response to the device state;

[0060] FIG. 6J is a tenth representation of an embodiment of a device configuration generated in response to the device state;

[0061] FIG. 7 ENTER FIGURES HERE;

[0062]

[0063]

[0064] In the appended figures, similar components and/or features may have the same reference label. Further, various components of the same type may be distinguished by following the reference label by a letter that distinguishes among the similar components. If only the first reference label is used in the specification, the description is applicable to any one of the similar components having the same first reference label irrespective of the second reference label.

DETAILED DESCRIPTION

[0065] Presented herein are embodiments of a device. The device can be a communications device, such as a cellular telephone, or other smart device. The device can include two screens that are oriented to provide several unique display configurations. Further, the device can receive user input in unique ways. The overall design and functionality of the device provides for an enhanced user experience making the device more useful and more efficient.

[0066] Mechanical Features:

[0067] Figs. 1A-1J illustrate a device 100 in accordance with embodiments of the present disclosure. As described in greater detail below, device 100 can be positioned in a number of different ways each of which provides different functionality to a user. The device 100 is a

multi-screen device that includes a primary screen 104 and a secondary screen 108, both of which are touch sensitive. In embodiments, the entire front surface of screens 104 and 108 may be touch sensitive and capable of receiving input by a user touching the front surface of the screens 104 and 108. Primary screen 104 includes touch sensitive display 110, which, in addition to being touch sensitive, also displays information to a user. Secondary screen 108 includes touch sensitive display 114, which also displays information to a user. In other embodiments, screens 104 and 108 may include more than one display area.

[0068] Primary screen 104 also includes a configurable area 112 that has been configured for specific inputs when the user touches portions of the configurable area 112. Secondary screen 108 also includes a configurable area 116 that has been configured for specific inputs. Areas 112a and 116a have been configured to receive a "back" input indicating that a user would like to view information previously displayed. Areas 112b and 116b have been configured to receive a "menu" input indicating that the user would like to view options from a menu. Areas 112c and 116c have been configured to receive a "home" input indicating that the user would like to view information associated with a "home" view. In other embodiments, areas 112a-c and 116a-c may be configured, in addition to the configurations described above, for other types of specific inputs including controlling features of device 100, some non-limiting examples including adjusting overall system power, adjusting the volume, adjusting the brightness, adjusting the vibration, selecting of displayed items (on either of screen 104 or 108), operating a camera, operating a microphone, and initiating/terminating of telephone calls. Also, in some embodiments, areas 112a-C and 116a-C may be configured for specific inputs depending upon the application running on device 100 and/or information displayed on touch sensitive displays 110 and/or 114.

[0069] In addition to touch sensing, primary screen 104 and secondary screen 108 may also include areas that receive input from a user without requiring the user to touch the display area of the screen. For example, primary screen 104 includes gesture capture area 120, and secondary screen 108 includes gesture capture area 124. These areas are able to receive input by recognizing gestures made by a user without the need for the user to actually touch the surface of the display area. In comparison to touch sensitive displays 110 and 114, the gesture capture areas 120 and 124 are commonly not capable of rendering a displayed image.

[0070] The two screens 104 and 108 are connected together with a hinge 128, shown clearly in Fig. 1C (illustrating a back view of device 100). Hinge 128, in the embodiment shown in Figs. 1A-1J, is a center hinge that connects screens 104 and 108 so that when the hinge is closed, screens 104 and 108 are juxtaposed (i.e., side-by-side) as shown in Fig. 1B (illustrating a front view of device 100). Hinge 128 can be opened to position the two screens 104 and 108 in different relative positions to each other. As described in greater detail below, the device 100 may have different functionalities depending on the relative positions of screens 104 and 108.

[0071] Fig. 1D illustrates the right side of device 100. As shown in Fig. 1D, secondary screen 108 also includes a card slot 132 and a port 136 on its side. Card slot 132 in embodiments, accommodates different types of cards including a subscriber identity module (SIM). Port 136 in embodiments is an input/output port (I/O port) that allows device 100 to be connected to other peripheral devices, such as a display, keyboard, or printing device. As can be appreciated, these are merely some examples and in other embodiments device 100 may include other slots and ports such as slots and ports for accommodating additional memory devices and/or for connecting other peripheral devices. Also shown in Fig. 1D is an audio jack 140 that accommodates a tip, ring, sleeve (TRS) connector for example to allow a user to utilize headphones or a headset.

[0072] Device 100 also includes a number of buttons 158. For example, Fig. 1E illustrates the lefr side of device 100. As shown in Fig. 1E, the side of primary screen 104 includes three buttons 144, 148, and 152, which can be configured for specific inputs. For example, buttons 144, 148, and 152 may be configured to, in combination or alone, control a number of aspects of device 100. Some non-limiting examples include overall system power, volume, brightness, vibration, selection of displayed items (on either of screen 104 or 108), a camera, a microphone, and initiation/termination of telephone calls. In some embodiments, instead of separate buttons two buttons may be combined into a rocker button. This arrangement is useful in situations where the buttons are configured to control features such as volume or brightness. In addition to buttons 144, 148, and 152, device 100 also includes a button 156, shown in Fig. 1F, which illustrates the top of device 100. In one embodiment, button 156 is configured as an on/off button used to control overall system power to device 100. In other embodiments, button 156 is configured to, in addition to or in lieu of controlling system power, control other aspects of device 100. In some embodiments, one or more of the buttons 144, 148, 152, and 156 are

capable of supporting different user commands. By way of example, a normal press has a duration commonly of less than about 1 second and resembles a quick tap. A medium press has a duration commonly of 1 second or more but less than about 12 seconds. A long press has a duration commonly of about 12 seconds or more. The function of the buttons is normally specific to the application that is currently in focus on the respective display 110 and 114. In a telephone application for instance and depending on the particular button, a normal, medium, or long press can mean end call, increase in call volume, decrease in call volume, and toggle microphone mute. In a camera or video application for instance and depending on the particular button, a normal, medium, or long press can mean increase zoom, decrease zoom, and take photograph or record video.

[0073] There are also a number of hardware components within device 100. As illustrated in Fig. 1C, device 100 includes a speaker 160 and a microphone 164. Device 100 also includes a camera 168 (Fig. 1B). Additionally, device 100 includes two position sensors 172A and 172B, which are used to determine the relative positions of screens 104 and 108. In one embodiment, position sensors 172A and 172B are Hall effect sensors. However, in other embodiments other sensors can be used in addition to or in lieu of the Hall effect sensors. An accelerometer 176 may also be included as part of device 100 to determine the orientation of the device 100 and/or the orientation of screens 104 and 108. Additional internal hardware components that may be included in device 100 are described below with respect to Fig. 2.

[0074] The overall design of device 100 allows it to provide additional functionality not available in other communication devices. Some of the functionality is based on the various positions and orientations that device 100 can have. As shown in Figs. 1B-1G, device 100 can be operated in an "open" position where screens 104 and 108 are juxtaposed. This position allows a large display area for displaying information to a user. When position sensors 172A and 172B determine that device 100 is in the open position, they can generate a signal that can be used to trigger different events such as displaying information on both screens 104 and 108. Additional events may be triggered if accelerometer 176 determines that device 100 is in a portrait position (Fig. 1B) as opposed to a landscape position (not shown).

[0075] In addition to the open position, device 100 may also have a "closed" position illustrated in Fig. 1H. Again, position sensors 172A and 172B can generate a signal indicating that device 100 is in the "closed" position. This can trigger an event that results in a change of

displayed information on screen 104 and/or 108. For example, device 100 may be programmed to stop displaying information on one of the screens, e.g., screen 108, since a user can only view one screen at a time when device 100 is in the "closed" position. In other embodiments, the signal generated by position sensors 172A and 172B, indicating that the device 100 is in the "closed" position, can trigger device 100 to answer an incoming telephone call. The "closed" position can also be a preferred position for utilizing the device 100 as a mobile phone.

[0076] Device 100 can also be used in an "easel" position which is illustrated in Fig. 1I. In the "easel" position, screens 104 and 108 are angled with respect to each other and facing outward with the edges of screens 104 and 108 substantially horizontal. In this position, device 100 can be configured to display information on both screens 104 and 108 to allow two users to simultaneously interact with device 100. When device 100 is in the "easel" position, sensors 172A and 172B generate a signal indicating that the screens 104 and 108 are positioned at an angle to each other, and the accelerometer 176 can generate a signal indicating that device 100 has been placed so that the edge of screens 104 and 108 are substantially horizontal. The signals can then be used in combination to generate events that trigger changes in the display of information on screens 104 and 108.

[0077] Fig. 1J illustrates device 100 in a "modified easel" position. In the "modified easel" position, one of screens 104 or 108 is used as a stand and is faced down on the surface of an object such as a table. This position provides a convenient way for information to be displayed to a user in landscape orientation. Similar to the easel position, when device 100 is in the "modified easel" position, position sensors 172A and 172B generate a signal indicating that the screens 104 and 108 are positioned at an angle to each other. The accelerometer 176 would generate a signal indicating that device 100 has been positioned so that one of screens 104 and 108 is faced downwardly and is substantially horizontal. The signals can then be used to generate events that trigger changes in the display of information of screens 104 and 108. For example, information may not be displayed on the screen that is face down since a user cannot see the screen.

[0078] Transitional states are also possible. When the position sensors 172A and B and/or accelerometer indicate that the screens are being closed or folded (from open), a closing transitional state is recognized. Conversely when the position sensors 172A and B indicate that the screens are being opened or folded (from closed), an opening transitional state is recognized.

The closing and opening transitional states are typically time-based, or have a maximum time duration from a sensed starting point. Normally, no user input is possible when one of the closing and opening states is in effect. In this manner, incidental user contact with a screen during the closing or opening function is not misinterpreted as user input. In embodiments, another transitional state is possible when the device 100 is closed. This additional transitional state allows the display to switch from one screen 104 to the second screen 108 when the device 100 is closed based on some user input, e.g., a double tap on the screen 110,114.

[0079] As can be appreciated, the description of device 100 is made for illustrative purposes only, and the embodiments are not limited to the specific mechanical features shown in Figs. 1A-1J and described above. In other embodiments, device 100 may include additional features, including one or more additional buttons, slots, display areas, hinges, and/or locking mechanisms. Additionally, in embodiments, the features described above may be located in different parts of device 100 and still provide similar functionality. Therefore, Figs. 1A-1J and the description provided above are nonlimiting.

[0080] Hardware Features:

[0081] Fig. 2 illustrates components of a device 100 in accordance with embodiments of the present disclosure. In general, the device 100 includes a primary screen 104 and a secondary screen 108. While the primary screen 104 and its components are normally enabled in both the opened and closed positions or states, the secondary screen 108 and its components are normally enabled in the opened state but disabled in the closed state. However, even when in the closed state a user or application triggered interrupt (such as in response to a phone application or camera application operation) can flip the active screen, or disable the primary screen 104 and enable the secondary screen 108, by a suitable command. Each screen 104, 108 can be touch sensitive and can include different operative areas. For example, a first operative area, within each touch sensitive screen 104 and 108, may comprise a touch sensitive display 110, 114. In general, the touch sensitive display 110, 114 may comprise a full color, touch sensitive display. A second area within each touch sensitive screen 104 and 108 may comprise a gesture capture region 120, 124. The gesture capture region 120, 124 may comprise an area or region that is outside of the touch sensitive display 110, 114 area, and that is capable of receiving input, for example in the form of gestures provided by a user. However, the gesture capture region 120, 124 does not include pixels that can perform a display function or capability.

[0082] A third region of the touch sensitive screens 104 and 108 may comprise a configurable area 112, 116. The configurable area 112, 116 is capable of receiving input and has display or limited display capabilities. In embodiments, the configurable area 112, 116 may present different input options to the user. For example, the configurable area 112, 116 may display buttons or other relatable items. Moreover, the identity of displayed buttons, or whether any buttons are displayed at all within the configurable area 112, 116 of a touch sensitive screen 104 or 108, may be determined from the context in which the device 100 is used and/or operated. In an exemplary embodiment, the touch sensitive screens 104 and 108 comprise liquid crystal display devices extending across at least those regions of the touch sensitive screens 104 and 108 that are capable of providing visual output to a user, and a capacitive input matrix over those regions of the touch sensitive screens 104 and 108 that are capable of receiving input from the user.

[0083] One or more display controllers 216a, 216b may be provided for controlling the operation of the touch sensitive screens 104 and 108, including input (touch sensing) and output (display) functions. In the exemplary embodiment illustrated in Fig. 2, a separate touch screen controller 216a or 216b is provided for each touch screen 104 and 108. In accordance with alternate embodiments, a common or shared touch screen controller 216 may be used to control each of the included touch sensitive screens 104 and 108. In accordance with still other embodiments, the functions of a touch screen controller 216 may be incorporated into other components, such as a processor 204.

[0084] The processor 204 may comprise a general purpose programmable processor or controller for executing application programming or instructions. In accordance with at least some embodiments, the processor 204 may include multiple processor cores, and/or implement multiple virtual processors. In accordance with still other embodiments, the processor 204 may include multiple physical processors. As a particular example, the processor 204 may comprise a specially configured application specific integrated circuit (ASIC) or other integrated circuit, a digital signal processor, a controller, a hardwired electronic or logic circuit, a programmable logic device or gate array, a special purpose computer, or the like. The processor 204 generally functions to run programming code or instructions implementing various functions of the device 100.

[0085] A communication device 100 may also include memory 208 for use in connection with the execution of application programming or instructions by the processor 204, and for the temporary or long term storage of program instructions and/or data. As examples, the memory 208 may comprise RAM, DRAM, SDRAM, or other solid state memory. Alternatively or in addition, data storage 212 may be provided. Like the memory 208, the data storage 212 may comprise a solid state memory device or devices. Alternatively or in addition, the data storage 212 may comprise a hard disk drive or other random access memory.

[0086] In support of communications functions or capabilities, the device 100 can include a cellular telephony module 228. As examples, the cellular telephony module 228 can comprise a GSM, CDMA, FDMA and/or analog cellular telephony transceiver capable of supporting voice, multimedia and/or data transfers over a cellular network. Alternatively or in addition, the device 100 can include an additional or other wireless communications module 232. As examples, the other wireless communications module 232 can comprise a Wi-Fi, BLUETOOTH TM, WiMax, infrared, or other wireless communications link. The cellular telephony module 228 and the other wireless communications module 232 can each be associated with a shared or a dedicated antenna 224.

[0087] A port interface 252 may be included. The port interface 252 may include proprietary or universal ports to support the interconnection of the device 100 to other devices or components, such as a dock, which may or may not include additional or different capabilities from those integral to the device 100. In addition to supporting an exchange of communication signals between the device 100 and another device or component, the docking port 136 and/or port interface 252 can support the supply of power to or from the device 100. The port interface 252 also comprises an intelligent element that comprises a docking module for controlling communications or other interactions between the device 100 and a connected device or component.

[0088] An input/output module 248 and associated ports may be included to support communications over wired networks or links, for example with other communication devices, server devices, and/or peripheral devices. Examples of an input/output module 248 include an Ethernet port, a Universal Serial Bus (USB) port, Institute of Electrical and Electronics Engineers (IEEE) 1394, or other interface.

[0089] An audio input/output interface/device(s) 244 can be included to provide analog audio to an interconnected speaker or other device, and to receive analog audio input from a connected microphone or other device. As an example, the audio input/output interface/device(s) 244 may comprise an associated amplifier and analog to digital converter. Alternatively or in addition, the device 100 can include an integrated audio input/output device 256 and/or an audio jack for interconnecting an external speaker or microphone. For example, an integrated speaker and an integrated microphone can be provided, to support near talk or speaker phone operations.

[0090] Hardware buttons 158 can be included for example for use in connection with certain control operations. Examples include a master power switch, volume control, etc., as described in conjunction with Figs. 1A through 1J. One or more image capture interfaces/devices 240, such as a camera, can be included for capturing still and/or video images. Alternatively or in addition, an image capture interface/device 240 can include a scanner or code reader. An image capture interface/device 240 can include or be associated with additional elements, such as a flash or other light source.

[0091] The device 100 can also include a global positioning system (GPS) receiver 236. In accordance with embodiments of the present invention, the GPS receiver 236 may further comprise a GPS module that is capable of providing absolute location information to other components of the device 100. An accelerometer(s) 176 may also be included. For example, in connection with the display of information to a user and/or other functions, a signal from the accelerometer 176 can be used to determine an orientation and/or format in which to display that information to the user.

[0092] Embodiments of the present invention can also include one or more position sensor(s) 172. The position sensor 172 can provide a signal indicating the position of the touch sensitive screens 104 and 108 relative to one another. This information can be provided as an input, for example to a user interface application, to determine an operating mode, characteristics of the touch sensitive displays 110, 114, and/or other device 100 operations. As examples, a screen position sensor 172 can comprise a series of Hall effect sensors, a multiple position switch, an optical switch, a Wheatstone bridge, a potentiometer, or other arrangement capable of providing a signal indicating of multiple relative positions the touch screens are in.

[0093] Communications between various components of the device 100 can be carried by one or more buses 222. In addition, power can be supplied to the components of the device 100 from

a power source and/or power control module 260. The power control module 260 can, for example, include a battery, an AC to DC converter, power control logic, and/or ports for interconnecting the device 100 to an external source of power.

[0094] Device State:

[0095] Figs. 3A and 3B represent illustrative states of device 100. While a number of illustrative states are shown, and transitions from a first state to a second state, it is to be appreciated that the illustrative state diagram may not encompass all possible states and/or all possible transitions from a first state to a second state. As illustrated in Fig. 3, the various arrows between the states (illustrated by the state represented in the circle) represent a physical change that occurs to the device 100, that is detected by one or more of hardware and software, the detection triggering one or more of a hardware and/or software interrupt that is used to control and/or manage one or more functions of device 100.

[0096] As illustrated in Fig. 3A, there are twelve exemplary "physical" states: closed 304, transition 308 (or opening transitional state), easel 312, modified easel 316, open 320, inbound/outbound call or communication 324, image/video capture 328, transition 332 (or closing transitional state), landscape 340, docked 336, docked 344 and landscape 348. Next to each illustrative state is a representation of the physical state of the device 100 with the exception of states 324 and 328, where the state is generally symbolized by the international icon for a telephone and the icon for a camera, respectfully.

[0097] In state 304, the device is in a closed state with the device 100 generally oriented in the portrait direction with the primary screen 104 and the secondary screen 108 back-to-back in different planes (see Fig. 1H). From the closed state, the device 100 can enter, for example, docked state 336, where the device 100 is coupled with a docking station, docking cable, or in general docked or associated with one or more other devices or peripherals, or the landscape state 340, where the device 100 is generally oriented with the primary screen 104 facing the user, and the primary screen 104 and the secondary screen 108 being back-to-back.

[0098] In the closed state, the device can also move to a transitional state where the device remains closed but the display is moved from one screen 104 to another screen 108 based on a user input, e.g., a double tap on the screen 110, 114. Still another embodiment includes a bilateral state. In the bilateral state, the device remains closed, but a single application displays at least one window on both the first display 110 and the second display 114. The windows

shown on the first and second display 110, 114 may be the same or different based on the application and the state of that application. For example, while acquiring an image with a camera, the device may display the view finder on the first display 110 and displays a preview for the photo subjects (full screen and mirrored left-to-right) on the second display 114.

[0099] In state 308, a transition state from the closed state 304 to the semi-open state or easel state 312, the device 100 is shown opening with the primary screen 104 and the secondary screen 108 being rotated around a point of axis coincidence with the hinge. Upon entering the easel state 312, the primary screen 104 and the secondary screen 108 are separated from one another such that, for example, the device 100 can sit in an easel-like configuration on a surface.

[0100] In state 316, known as the modified easel position, the device 100 has the primary screen 104 and the secondary screen 108 in a similar relative relationship to one another as in the easel state 312, with the difference being one of the primary screen 104 or the secondary screen 108 are placed on a surface as shown.

[0101] State 320 is the open state where the primary screen 104 and the secondary screen 108 are generally on the same plane. From the open state, the device 100 can transition to the docked state 344 or the open landscape state 348. In the open state 320, the primary screen 104 and the secondary screen 108 are generally in the portrait-like orientation while in landscaped state 348 the primary screen 104 and the secondary screen 108 are generally in a landscape-like orientation.

[0102] State 324 is illustrative of a communication state, such as when an inbound or outbound call is being received or placed, respectively, by the device 100. While not illustrated for clarity, it should be appreciated the device 100 can transition to the inbound/outbound call state 324 from any state illustrated in Fig. 3. In a similar manner, the image/video capture state 328 can be entered into from any other state in Fig. 3, with the image/video capture state 328 allowing the device 100 to take one or more images via a camera and/or videos with a video capture device 240.

[0103] Transition state 322 illustratively shows primary screen 104 and the secondary screen 108 being closed upon one another for entry into, for example, the closed state 304.

[0104] Fig. 3B illustrates, with reference to the key, the inputs that are received to detect a transition from a first state to a second state. In Fig. 3B, various combinations of states are shown with in general, a portion of the columns being directed toward a portrait state 352, a

landscape state 356, and a portion of the rows being directed to portrait state 360 and landscape state 364.

[0105] In Fig. 3B, the Key indicates that "H" represents an input from one or more Hall Effect sensors, "A" represents an input from one or more accelerometers, "T" represents an input from a timer, "P" represents a communications trigger input and "I" represents an image and/or video capture request input. Thus, in the center portion 376 of the chart, an input, or combination of inputs, are shown that represent how the device 100 detects a transition from a first physical state to a second physical state.

[0106] As discussed, in the center portion of the chart 376, the inputs that are received enable the detection of a transition from, for example, a portrait open state to a landscape easel state - shown in bold – "HAT." For this exemplary transition from the portrait open to the landscape easel state, a Hall Effect sensor ("H"), an accelerometer ("A") and a timer ("T") input may be needed. The timer input can be derived from, for example, a clock associated with the processor.

[0107] In addition to the portrait and landscape states, a docked state 368 is also shown that is triggered based on the receipt of a docking signal 372. As discussed above and in relation to Fig. 3, the docking signal can be triggered by the association of the device 100 with one or more other device 100s, accessories, peripherals, smart docks, or the like.

[0108] User Interaction:

[0109] Figs. 4A through 4H depict various graphical representations of gesture inputs that may be recognized by the screens 104, 108. The gestures may be performed not only by a user's body part, such as a digit, but also by other devices, such as a stylus, that may be sensed by the contact sensing portion(s) of a screen 104, 108. In general, gestures are interpreted differently, based on where the gestures are performed (either directly on the display 110, 114 or in the gesture capture region 120, 124). For example, gestures in the display 110,114 may be directed to a desktop or application, and gestures in the gesture capture region 120, 124 may be interpreted as for the system.

[0110] With reference to Figs. 4A-4H, a first type of gesture, a touch gesture 420, is substantially stationary on the screen 104,108 for a selected length of time. A circle 428 represents a touch or other contact type received at particular location of a contact sensing portion of the screen. The circle 428 may include a border 432, the thickness of which indicates a length of time that the contact is held substantially stationary at the contact location. For

instance, a tap 420 (or short press) has a thinner border 432a than the border 432b for a long press 424 (or for a normal press). The long press 424 may involve a contact that remains substantially stationary on the screen for longer time period than that of a tap 420. As will be appreciated, differently defined gestures may be registered depending upon the length of time that the touch remains stationary prior to contact cessation or movement on the screen.

[0111] With reference to Fig. 4C, a drag gesture 400 on the screen 104,108 is an initial contact (represented by circle 428) with contact movement 436 in a selected direction. The initial contact 428 may remain stationary on the screen 104,108 for a certain amount of time represented by the border 432. The drag gesture typically requires the user to contact an icon, window, or other displayed image at a first location followed by movement of the contact in a drag direction to a new second location desired for the selected displayed image. The contact movement need not be in a straight line but have any path of movement so long as the contact is substantially continuous from the first to the second locations.

[0112] With reference to Fig. 4D, a flick gesture 404 on the screen 104,108 is an initial contact (represented by circle 428) with truncated contact movement 436 (relative to a drag gesture) in a selected direction. In embodiments, a flick has a higher exit velocity for the last movement in the gesture compared to the drag gesture. The flick gesture can, for instance, be a finger snap following initial contact. Compared to a drag gesture, a flick gesture generally does not require continual contact with the screen 104,108 from the first location of a displayed image to a predetermined second location. The contacted displayed image is moved by the flick gesture in the direction of the flick gesture to the predetermined second location. Although both gestures commonly can move a displayed image from a first location to a second location, the temporal duration and distance of travel of the contact on the screen is generally less for a flick than for a drag gesture.

[0113] With reference to Fig. 4E, a pinch gesture 408 on the screen 104,108 is depicted. The pinch gesture 408 may be initiated by a first contact 428 to the screen 104,108 by, for example, a first digit and a second contact 428b to the screen 104,108 by, for example, a second digit. The first and second contacts 428a,b may be detected by a common contact sensing portion of a common screen 104,108, by different contact sensing portions of a common screen 104 or108, or by different contact sensing portions of different screens. The first contact 428a is held for a first amount of time, as represented by the border 432a, and the second contact 428b is held for a

second amount of time, as represented by the border 432b. The first and second amounts of time are generally substantially the same, and the first and second contacts 428 a, b generally occur substantially simultaneously. The first and second contacts 428 a, b generally also include corresponding first and second contact movements 436 a, b, respectively. The first and second contact movements 436 a, b are generally in opposing directions. Stated another way, the first contact movement 436a is towards the second contact 436b, and the second contact movement 436b is towards the first contact 436a. More simply stated, the pinch gesture 408 may be accomplished by a user's digits touching the screen 104,108 in a pinching motion.

[0114] With reference to Fig. 4F, a spread gesture 410 on the screen 104,108 is depicted. The spread gesture 410 may be initiated by a first contact 428a to the screen 104,108 by, for example, a first digit and a second contact 428b to the screen 104,108 by, for example, a second digit. The first and second contacts 428a,b may be detected by a common contact sensing portion of a common screen 104,108, by different contact sensing portions of a common screen 104,108, or by different contact sensing portions of different screens. The first contact 428a is held for a first amount of time, as represented by the border 432a, and the second contact 428b is held for a second amount of time, as represented by the border 432b. The first and second amounts of time are generally substantially the same, and the first and second contacts 428 a, b generally occur substantially simultaneously. The first and second contacts 428 a, b generally also include corresponding first and second contact movements 436a, b, respectively. The first and second contact movements 436 a, b are generally in a common direction. Stated another way, the first and second contact movements 436 a, b are away from the first and second contacts 428a, b. More simply stated, the spread gesture 410 may be accomplished by a user's digits touching the screen 104,108 in a spreading motion.

[0115] The above gestures may be combined in any manner, such as those shown by Figs. 4G and 4H, to produce a determined functional result. For example, in Fig. 4G a tap gesture 420 is combined with a drag or flick gesture 412 in a direction away from the tap gesture 420. In Fig. 4H, a tap gesture 420 is combined with a drag or flick gesture 412 in a direction towards the tap gesture 420.

[0116] The functional result of receiving a gesture can vary depending on a number of factors, including a state of the device 100, display 110, 114, or screen 104, 108, a context associated with the gesture, or sensed location of the gesture. The state of the device commonly refers to

one or more of a configuration of the device 100, a display orientation, and user and other inputs received by the device 100. Context commonly refers to one or more of the particular application(s) selected by the gesture and the portion(s) of the application currently executing, whether the application is a single- or multi-screen application, and whether the application is a multi-screen application displaying one or more windows in one or more screens or in one or more stacks. Sensed location of the gesture commonly refers to whether the sensed set(s) of gesture location coordinates are on a touch sensitive display 110, 114 or a gesture capture region 120, 124, whether the sensed set(s) of gesture location coordinates are associated with a common or different display or screen 104,108, and/or what portion of the gesture capture region contains the sensed set(s) of gesture location coordinates.

[0117] A tap, when received by an a touch sensitive display 110, 114, can be used, for instance, to select an icon to initiate or terminate execution of a corresponding application, to maximize or minimize a window, to reorder windows in a stack, and to provide user input such as by keyboard display or other displayed image. A drag, when received by a touch sensitive display 110, 114, can be used, for instance, to relocate an icon or window to a desired location within a display, to reorder a stack on a display, or to span both displays (such that the selected window occupies a portion of each display simultaneously). A flick, when received by a touch sensitive display 110, 114 or a gesture capture region 120, 124, can be used to relocate a window from a first display to a second display or to span both displays (such that the selected window occupies a portion of each display simultaneously). Unlike the drag gesture, however, the flick gesture is generally not used to move the displayed image to a specific user-selected location but to a default location that is not configurable by the user.

[0118] The pinch gesture, when received by a touch sensitive display 110, 114 or a gesture capture region 120, 124, can be used to minimize or otherwise increase the displayed area or size of a window (typically when received entirely by a common display), to switch windows displayed at the top of the stack on each display to the top of the stack of the other display (typically when received by different displays or screens), or to display an application manager (a "pop-up window" that displays the windows in the stack). The spread gesture, when received by a touch sensitive display 110, 114 or a gesture capture region 120, 124, can be used to maximize or otherwise decrease the displayed area or size of a window, to switch windows displayed at the top of the stack on each display to the top of the stack of the other display

(typically when received by different displays or screens), or to display an application manager (typically when received by an off-screen gesture capture region on the same or different screens).

[0119] The combined gestures of Fig. 4G, when received by a common display capture region in a common display or screen 104,108, can be used to hold a first window stack location in a first stack constant for a display receiving the gesture while reordering a second window stack location in a second window stack to include a window in the display receiving the gesture. The combined gestures of Fig. 4H, when received by different display capture regions in a common display or screen 104,108 or in different displays or screens, can be used to hold a first window stack location in a first window stack constant for a display receiving the tap part of the gesture while reordering a second window stack location in a second window stack to include a window in the display receiving the flick or drag gesture. Although specific gestures and gesture capture regions in the preceding examples have been associated with corresponding sets of functional results, it is to be appreciated that these associations can be redefined in any manner to produce differing associations between gestures and/or gesture capture regions and/or functional results.

[0120] Firmware and Software:

[0121] The memory 508 may store and the processor 504 may execute one or more software components. These components can include at least one operating system (OS) 516, an application manager 562, a desktop 566, and/or one or more applications 564a and/or 564b from an application store 560. The OS 516 can include a framework 520, one or more frame buffers 548, one or more drivers 512, previously described in conjunction with Fig. 2, and/or a kernel 518. The OS 516 can be any software, consisting of programs and data, which manages computer hardware resources and provides common services for the execution of various applications 564. The OS 516 can be any operating system and, at least in some embodiments, dedicated to mobile devices, including, but not limited to, Linux, ANDROID TM, iPhone OS (IOS TM), WINDOWS PHONE 7 TM, etc. The OS 516 is operable to provide functionality to the phone by executing one or more operations, as described herein.

[0122] The applications 564 can be any higher level software that executes particular functionality for the user. Applications 564 can include programs such as email clients, web browsers, texting applications, games, media players, office suites, etc. The applications 564 can be stored in an application store 560, which may represent any memory or data storage, and the

management software associated therewith, for storing the applications 564. Once executed, the applications 564 may be run in a different area of memory 508.

[0123] The framework 520 may be any sofrware or data that allows the multiple tasks running on the device to interact. In embodiments, at least portions of the framework 520 and the discrete components described hereinafrer may be considered part of the OS 516 or an application 564. However, these portions will be described as part of the framework 520, but those components are not so limited. The framework 520 can include, but is not limited to, a Multi-Display Management (MDM) module 524, a Surface Cache module 528, a Window Management module 532, an Input Management module 536, a Task Management module 540, an Application Model Manager 542, a Display Controller, one or more frame buffers 548, a task stack 552, one or more window stacks 550 (which is a logical arrangement of windows and/or desktops in a display area), and/or an event buffer 556.

[0124] The MDM module 524 includes one or more modules that are operable to manage the display of applications or other data on the screens of the device. An embodiment of the MDM module 524 is described in conjunction with Fig. 5B. In embodiments, the MDM module 524 receives inputs from the other OS 516 components, such as, the drivers 512, and from the applications 564 to determine continually the state of the device 100. The inputs assist the MDM module 524 in determining how to configure and allocate the displays according to the application's preferences and requirements, and the user's actions. Once a determination for display configurations is made, the MDM module 524 can bind the applications 564 to a display. The configuration may then be provided to one or more other components to generate a window with a display.

[0125] The Surface Cache module 528 includes any memory or storage and the software associated therewith to store or cache one or more images of windows. A series of active and/or non-active windows (or other display objects, such as, a desktop display) can be associated with each display. An active window (or other display object) is currently displayed. A non-active windows (or other display objects) were opened and, at some time, displayed but are now not displayed. To enhance the user experience, before a window transitions from an active state to an inactive state, a "screen shot" of a last generated image of the window (or other display object) can be stored. The Surface Cache module 528 may be operable to store a bitmap of the last active image of a window (or other display object) not currently displayed. Thus, the

Surface Cache module 528 stores the images of non-active windows (or other display objects) in a data store.

[0126] In embodiments, the Window Management module 532 is operable to manage the windows (or other display objects) that are active or not active on each of the displays. The Window Management module 532, based on information from the MDM module 524, the OS 516, or other components, determines when a window (or other display object) is visible or not active. The Window Management module 532 may then put a non-visible window (or other display object) in a "not active state" and, in conjunction with the Task Management module Task Management 540 suspends the application's operation. Further, the Window Management module 532 may assign, through collaborative interaction with the MDM module 524, a display identifier to the window (or other display object) or manage one or more other items of data associated with the window (or other display object). The Window Management module 532 may also provide the stored information to the application 564, the Task Management module 540, or other components interacting with or associated with the window (or other display object). The Window Management module 532 can also associate an input task with a window based on window focus and display coordinates within the motion space.

[0127] The Input Management module 536 is operable to manage events that occur with the device. An event is any input into the window environment, for example, a user interface interactions with a user. The Input Management module 536 receives the events and logically stores the events in an event buffer 556. Events can include such user interface interactions as a "down event," which occurs when a screen 104, 108 receives a touch signal from a user, a "move event," which occurs when the screen 104, 108 determines that a user's finger is moving across a screen(s), an "up event, which occurs when the screen 104, 108 determines that the user has stopped touching the screen 104, 108, etc. These events are received, stored, and forwarded to other modules by the Input Management module 536. The Input Management module 536 may also map screen inputs to a motion space which is the culmination of all physical and virtual display available on the device.

[0128] The motion space is a virtualized space that includes all touch sensitive displays 110,114 "tiled" together to mimic the physical dimensions of the device 100. For example, when the device 100 is unfolded, the motion space size may be 960x800, which may be the number of pixels in the combined display area for both touch sensitive displays 110, 114. If a user touches

on a first touch sensitive display 110 on location (40, 40), a full screen window can receive touch event with location (40, 40). If a user touches on a second touch sensitive display 114, with location (40, 40), the full screen window can receive touch event with location (520, 40), because the second touch sensitive display 114 is on the right side of the first touch sensitive display 110, so the device 100 can offset the touch by the first touch sensitive display's 110 width, which is 480 pixels. When a hardware event occurs with location info from a driver 512, the framework 520 can up-scale the physical location to the motion space because the location of the event may be different based on the device orientation and state. The motion space may be as described in U.S. Patent Application No. 13/187,026, filed July 20, 2011, entitled "Systems and Methods for Receiving Gesture Inputs Spanning Multiple Input Devices," which is hereby incorporated by reference in its entirety for all that it teaches and for all purposes.

[0129] A task can be an application and a sub-task can be an application component that provides a window with which users can interact to do something, such as dial the phone, take a photo, send an email, or view a map. Each task may be given a window in which to draw a user interface. The window typically fills a display (for example, touch sensitive display 110,114), but may be smaller than the display 110,114 and float on top of other windows. An application usually consists of multiple sub-tasks that are loosely bound to each other. Typically, one task in an application is specified as the "main" task, which is presented to the user when launching the application for the first time. Each task can then start another task or sub-task to perform different actions.

[0130] The Task Management module 540 is operable to manage the operation of one or more applications 564 that may be executed by the device. Thus, the Task Management module 540 can receive signals to launch, suspend, terminate, etc. an application or application sub-tasks stored in the application store 560. The Task Management module 540 may then instantiate one or more tasks or sub-tasks of the application 564 to begin operation of the application 564. Further, the Task Management Module 540 may launch, suspend, or terminate a task or sub-task as a result of user input or as a result of a signal from a collaborating framework 520 component. The Task Management Module 540 is responsible for managing the lifecycle of applications (tasks and sub-task) from when the application is launched to when the application is terminated.

[0131] The processing of the Task Management Module 540 is facilitated by a task stack 552, which is a logical structure associated with the Task Management Module 540. The task stack

552 maintains the state of all tasks and sub-tasks on the device 100. When some component of the operating system 516 requires a task or sub-task to transition in its lifecycle, the OS 516 component can notify the Task Management Module 540. The Task Management Module 540 may then locate the task or sub-task, using identification information, in the task stack 552, and send a signal to the task or sub-task indicating what kind of lifecycle transition the task needs to execute. Informing the task or sub-task of the transition allows the task or sub-task to prepare for the lifecycle state transition. The Task Management Module 540 can then execute the state transition for the task or sub-task. In embodiments, the state transition may entail triggering the OS kernel 518 to terminate the task when termination is required.

[0132] Further, the Task Management module 540 may suspend the application 564 based on information from the Window Management Module 532. Suspending the application 564 may maintain application data in memory but may limit or stop the application 564 from rendering a window or user interface. Once the application becomes active again, the Task Management module 540 can again trigger the application to render its user interface. In embodiments, if a task is suspended, the task may save the task's state in case the task is terminated. In the suspended state, the application task may not receive input because the application window is not visible to the user.

[0133] The frame buffer 548 is a logical structure(s) used to render the user interface. The frame buffer 548 can be created and destroyed by the OS kernel 518. However, the Display Controller 544 can write the image data, for the visible windows, into the frame buffer 548. A frame buffer 548 can be associated with one screen or multiple screens. The association of a frame buffer 548 with a screen can be controlled dynamically by interaction with the OS kernel 518. A composite display may be created by associating multiple screens with a single frame buffer 548. Graphical data used to render an application's window user interface may then be written to the single frame buffer 548, for the composite display, which is output to the multiple screens 104,108. The Display Controller 544 can direct an application's user interface to a portion of the frame buffer 548 that is mapped to a particular display 110,114, thus, displaying the user interface on only one screen 104 or 108. The Display Controller 544 can extend the control over user interfaces to multiple applications, controlling the user interfaces for as many displays as are associated with a frame buffer 548 or a portion thereof. This approach

compensates for the multiple physical screens 104,108 that are in use by the software component above the Display Controller 544.

[0134] The Application Manager 562 is an application that provides a presentation layer for the window environment. Thus, the Application Manager 562 provides the graphical model for rendering by the Task Management Module 540. Likewise, the Desktop 566 provides the presentation layer for the Application Store 560. Thus, the desktop provides a graphical model of a surface having selectable application icons for the Applications 564 in the Application Store 560 that can be provided to the Window Management Module 556 for rendering.

[0135] Further, the framework can include an Application Model Manager (AMM) 542. The Application Manager 562 may interface with the AMM 542. In embodiments, the AMM 542 receives state change information from the device 100 regarding the state of applications (which are running or suspended). The AMM 542 can associate bit map images from the Surface Cache Module 528 to the tasks that are alive (running or suspended). Further, the AMM 542 can convert the logical window stack maintained in the Task Manager Module 540 to a linear ("film strip" or "deck of cards") organization that the user perceives when the using the off gesture capture area 120 to sort through the windows. Further, the AMM 542 may provide a list of executing applications to the Application Manager 562.

[0136] An embodiment of the MDM module 524 is shown in Fig. 5B. The MDM module 524 is operable to determine the state of the environment for the device, including, but not limited to, the orientation of the device, whether the device 100 is opened or closed, what applications 564 are executing, how the applications 564 are to be displayed, what actions the user is conducting, the tasks being displayed, etc. To configure the display, the MDM module 524 interprets these environmental factors and determines a display configuration, as described in conjunction with Figs. 6A-6J. Then, the MDM module 524 can bind the applications 564 or other device components to the displays. The configuration may then be sent to the Display Controller 544 and/or the other components within the OS 516 to generate the display. The MDM module 524 can include one or more of, but is not limited to, a Display Configuration Module 568, a Preferences Module 572, a Device State Module 574, a Gesture Module 576, a Requirements Module 580, an Event Module 584, and/or a Binding Module 588.

[0137] The Display Configuration Module 568 determines the layout for the display. In embodiments, the Display Configuration Module 568 can determine the environmental factors.

The environmental factors may be received from one or more other MDM modules 524 or from other sources. The Display Configuration Module 568 can then determine from the list of factors the best configuration for the display. Some embodiments of the possible configurations and the factors associated therewith are described in conjunction with Figs. 6A-6F.

[0138] The Preferences Module 572 is operable to determine display preferences for an application 564 or other component. For example, an application can have a preference for Single or Dual displays. The Preferences Module 572 can determine an application's display preference (e.g., by inspecting the application's preference settings) and may allow the application 564 to change to a mode (e.g., single screen, dual screen, max, etc.) if the device 100 is in a state that can accommodate the preferred mode. However, some user interface policies may disallow a mode even if the mode is available. As the configuration of the device changes, the preferences may be reviewed to determine if a better display configuration can be achieved for an application 564.

[0139] The Device State Module 574 is operable to determine or receive the state of the device. The state of the device can be as described in conjunction with Figs. 3A and 3B. The state of the device can be used by the Display Configuration Module 568 to determine the configuration for the display. As such, the Device State Module 574 may receive inputs and interpret the state of the device. The state information is then provided to the Display Configuration Module 568.

[0140] The Gesture Module 576 is shown as part of the MDM module 524, but, in embodiments, the Gesture module 576 may be a separate Framework 520 component that is separate from the MDM module 524. In embodiments, the Gesture Module 576 is operable to determine if the user is conducting any actions on any part of the user interface. In alternative embodiments, the Gesture Module 576 receives user interface actions from the configurable area 112,116 only. The Gesture Module 576 can receive touch events that occur on the configurable area 112,116 (or possibly other user interface areas) by way of the Input Management Module 536 and may interpret the touch events (using direction, speed, distance, duration, and various other parameters) to determine what kind of gesture the user is performing. When a gesture is interpreted, the Gesture Module 576 can initiate the processing of the gesture and, by collaborating with other Framework 520 components, can manage the required window animation. The Gesture Module 576 collaborates with the Application Model Manager 542 to

collect state information with respect to which applications are running (active or paused) and the order in which applications must appear when a user gesture is performed. The Gesture Module 576 may also receive references to bitmaps (from the Surface Cache Module 528) and live windows so that when a gesture occurs it can instruct the Display Controller 544 how to move the window(s) across the display 110,114. Thus, suspended applications may appear to be running when those windows are moved across the display 110,114.

[0141] Further, the Gesture Module 576 can receive task information either from the Task Manage Module 540 or the Input Management module 536. The gestures may be as defined in conjunction with Figs. 4A through 4H. For example, moving a window causes the display to render a series of display frames that illustrate the window moving. The gesture associated with such user interface interaction can be received and interpreted by the Gesture Module 576. The information about the user gesture is then sent to the Task Management Module 540 to modify the display binding of the task.

[0142] The Requirements Module 580, similar to the Preferences Module 572, is operable to determine display requirements for an application 564 or other component. An application can have a set display requirement that must be observed. Some applications require a particular display orientation. For example, the application "Angry Birds" can only be displayed in landscape orientation. This type of display requirement can be determined or received, by the Requirements Module 580. As the orientation of the device changes, the Requirements Module 580 can reassert the display requirements for the application 564. The Display Configuration Module 568 can generate a display configuration that is in accordance with the application display requirements, as provided by the Requirements Module 580.

[0143] The Event Module 584, similar to the Gesture Module 576, is operable to determine one or more events occurring with an application or other component that can affect the user interface. Thus, the Event Module 584 can receive event information either from the event buffer 556 or the Task Management module 540. These events can change how the tasks are bound to the displays. The Event Module 584 can collect state change information from other Framework 520 components and act upon that state change information. In an example, when the phone is opened or closed or when an orientation change has occurred, a new message may be rendered in a secondary screen. The state change based on the event can be received and

interpreted by the Event Module 584. The information about the events then may be sent to the Display Configuration Module 568 to modify the configuration of the display.

[0144] The Binding Module 588 is operable to bind the applications 564 or the other components to the configuration determined by the Display Configuration Module 568. A binding associates, in memory, the display configuration for each application with the display and mode of the application. Thus, the Binding Module 588 can associate an application with a display configuration for the application (e.g. landscape, portrait, multi-screen, etc.). Then, the Binding Module 588 may assign a display identifier to the display. The display identifier associated the application with a particular display of the device 100. This binding is then stored and provided to the Display Controller 544, the other components of the OS 516, or other components to properly render the display. The binding is dynamic and can change or be updated based on configuration changes associated with events, gestures, state changes, application preferences or requirements, etc.

[0145] User Interface Configurations:

[0146] With reference now to Figs. 6A-J, various types of output configurations made possible by the device 100 will be described hereinafrer.

[0147] Figs. 6A and 6B depict two different output configurations of the device 100 being in a first state. Specifically, Fig. 6A depicts the device 100 being in a closed portrait state 304 where the data is displayed on the primary screen 104. In this example, the device 100 displays data via the touch sensitive display 110 in a first portrait configuration 604. As can be appreciated, the first portrait configuration 604 may only display a desktop or operating system home screen. Alternatively, one or more windows may be presented in a portrait orientation while the device 100 is displaying data in the first portrait configuration 604.

[0148] Fig. 6B depicts the device 100 still being in the closed portrait state 304, but instead data is displayed on the secondary screen 108. In this example, the device 100 displays data via the touch sensitive display 114 in a second portrait configuration 608.

[0149] It may be possible to display similar or different data in either the first or second portrait configuration 604, 608. It may also be possible to transition between the first portrait configuration 604 and second portrait configuration 608 by providing the device 100 a user gesture (e.g., a double tap gesture), a menu selection, or other means. Other suitable gestures may also be employed to transition between configurations. Furthermore, it may also be possible

to transition the device 100 from the first or second portrait configuration 604, 608 to any other configuration described herein depending upon which state the device 100 is moved.

[0150] An alternative output configuration may be accommodated by the device 100 being in a second state. Specifically, Fig. 6C depicts a third portrait configuration where data is displayed simultaneously on both the primary screen 104 and the secondary screen 108. The third portrait configuration may be referred to as a Dual-Portrait (PD) output configuration. In the PD output configuration, the touch sensitive display 110 of the primary screen 104 depicts data in the first portrait configuration 604 while the touch sensitive display 114 of the secondary screen 108 depicts data in the second portrait configuration 608. The simultaneous presentation of the first portrait configuration 604 and the second portrait configuration 608 may occur when the device 100 is in an open portrait state 320. In this configuration, the device 100 may display one application window in one display 110 or 114, two application windows (one in each display 110 and 114), one application window and one desktop, or one desktop. Other configurations may be possible. It should be appreciated that it may also be possible to transition the device 100 from the simultaneous display of configurations 604, 608 to any other configuration described herein depending upon which state the device 100 is moved. Furthermore, while in this state, an application's display preference may place the device into bilateral mode, in which both displays are active to display different windows in the same application. For example, a Camera application may display a viewfinder and controls on one side, while the other side displays a mirrored preview that can be seen by the photo subjects. Games involving simultaneous play by two players may also take advantage of bilateral mode.

[0151] Figs. 6D and 6E depicts two further output configurations of the device 100 being in a third state. Specifically, Fig. 6D depicts the device 100 being in a closed landscape state 340 where the data is displayed on the primary screen 104. In this example, the device 100 displays data via the touch sensitive display 110 in a first landscape configuration 612. Much like the other configurations described herein, the first landscape configuration 612 may display a desktop, a home screen, one or more windows displaying application data, or the like.

[0152] Fig. 6E depicts the device 100 still being in the closed landscape state 340, but instead data is displayed on the secondary screen 108. In this example, the device 100 displays data via the touch sensitive display 114 in a second landscape configuration 616. It may be possible to display similar or different data in either the first or second portrait configuration 612, 616. It

may also be possible to transition between the first landscape configuration 612 and second landscape configuration 616 by providing the device 100 with one or both of a twist and tap gesture or a flip and slide gesture. Other suitable gestures may also be employed to transition between configurations. Furthermore, it may also be possible to transition the device 100 from the first or second landscape configuration 612, 616 to any other configuration described herein depending upon which state the device 100 is moved.

[0153] Fig. 6F depicts a third landscape configuration where data is displayed simultaneously on both the primary screen 104 and the secondary screen 108. The third landscape configuration may be referred to as a Dual-Landscape (LD) output configuration. In the LD output configuration, the touch sensitive display 110 of the primary screen 104 depicts data in the first landscape configuration 612 while the touch sensitive display 114 of the secondary screen 108 depicts data in the second landscape configuration 616. The simultaneous presentation of the first landscape configuration 612 and the second landscape configuration 616 may occur when the device 100 is in an open landscape state 340. It should be appreciated that it may also be possible to transition the device 100 from the simultaneous display of configurations 612, 616 to any other configuration described herein depending upon which state the device 100 is moved.

[0154] Figs. 6G and 6H depict two views of a device 100 being in yet another state. Specifically, the device 100 is depicted as being in an easel state 312. Fig. 6G shows that a first easel output configuration 618 may be displayed on the touch sensitive display 110. Fig. 6H shows that a second easel output configuration 620 may be displayed on the touch sensitive display 114. The device 100 may be configured to depict either the first easel output configuration 618 or the second easel output configuration 620 individually. Alternatively, both the easel output configurations 618, 620 may be presented simultaneously. In some embodiments, the easel output configurations 618, 620 may be similar or identical to the landscape output configurations 612, 616. The device 100 may also be configured to display one or both of the easel output configurations 618, 620 while in a modified easel state 316. It should be appreciated that simultaneous utilization of the easel output configurations 618, 620 may facilitate two-person games (e.g., Battleship®, chess, checkers, etc.), multi-user conferences where two or more users share the same device 100, and other applications. As can be appreciated, it may also be possible to transition the device 100 from the display of one or both

configurations 618, 620 to any other configuration described herein depending upon which state the device 100 is moved.

[0155] Fig. 6I depicts yet another output configuration that may be accommodated while the device 100 is in an open portrait state 320. Specifically, the device 100 may be configured to present a single continuous image across both touch sensitive displays 110, 114 in a portrait configuration referred to herein as a Portrait-Max (PMax) configuration 624. In this configuration, data (e.g., a single image, application, window, icon, video, etc.) may be split and displayed partially on one of the touch sensitive displays while the other portion of the data is displayed on the other touch sensitive display. The Pmax configuration 624 may facilitate a larger display and/or better resolution for displaying a particular image on the device 100. Similar to other output configurations, it may be possible to transition the device 100 from the Pmax configuration 624 to any other output configuration described herein depending upon which state the device 100 is moved.

[0156] Fig. 6J depicts still another output configuration that may be accommodated while the device 100 is in an open landscape state 348. Specifically, the device 100 may be configured to present a single continuous image across both touch sensitive displays 110, 114 in a landscape configuration referred to herein as a Landscape-Max (LMax) configuration 628. In this configuration, data (e.g., a single image, application, window, icon, video, etc.) may be split and displayed partially on one of the touch sensitive displays while the other portion of the data is displayed on the other touch sensitive display. The Lmax configuration 628 may facilitate a larger display and/or better resolution for displaying a particular image on the device 100. Similar to other output configurations, it may be possible to transition the device 100 from the Lmax configuration 628 to any other output configuration described herein depending upon which state the device 100 is moved.

[0157] INSERT YOUR DESCRIPTION HERE

[0158] The exemplary systems and methods of this disclosure have been described in relation to _____. However, to avoid unnecessarily obscuring the present disclosure, the preceding description omits a number of known structures and devices. This omission is not to be construed as a limitation of the scopes of the claims. Specific details are set forth to provide an understanding of the present disclosure. It should however be appreciated

that the present disclosure may be practiced in a variety of ways beyond the specific detail set forth herein.

[0159] Furthermore, while the exemplary aspects, embodiments, and/or configurations illustrated herein show the various components of the system collocated, certain components of the system can be located remotely, at distant portions of a distributed network, such as a LAN and/or the Internet, or within a dedicated system. Thus, it should be appreciated, that the components of the system can be combined in to one or more devices, such as a _____, or collocated on a particular node of a distributed network, such as an analog and/or digital telecommunications network, a packet-switch network, or a circuit-switched network. It will be appreciated from the preceding description, and for reasons of computational efficiency, that the components of the system can be arranged at any location within a distributed network of components without affecting the operation of the system. For example, the various components can be located in a switch such as a PBX and media server, gateway, in one or more communications devices, at one or more users' premises, or some combination thereof. Similarly, one or more functional portions of the system could be distributed between a telecommunications device(s) and an associated computing device.

[0160] Furthermore, it should be appreciated that the various links connecting the elements can be wired or wireless links, or any combination thereof, or any other known or later developed element(s) that is capable of supplying and/or communicating data to and from the connected elements. These wired or wireless links can also be secure links and may be capable of communicating encrypted information. Transmission media used as links, for example, can be any suitable carrier for electrical signals, including coaxial cables, copper wire and fiber optics, and may take the form of acoustic or light waves, such as those generated during radio-wave and infra-red data communications.

[0161] Also, while the flowcharts have been discussed and illustrated in relation to a particular sequence of events, it should be appreciated that changes, additions, and omissions to this sequence can occur without materially affecting the operation of the disclosed embodiments, configuration, and aspects.

[0162] A number of variations and modifications of the disclosure can be used. It would be possible to provide for some features of the disclosure without providing others.

[0163] For example in one alternative embodiment, _____.

[0164] In another alternative embodiment, _____.

[0165] In yet another embodiment, the systems and methods of this disclosure can be implemented in conjunction with a special purpose computer, a programmed microprocessor or microcontroller and peripheral integrated circuit element(s), an ASIC or other integrated circuit, a digital signal processor, a hard-wired electronic or logic circuit such as discrete element circuit, a programmable logic device or gate array such as PLD, PLA, FPGA, PAL, special purpose computer, any comparable means, or the like. In general, any device(s) or means capable of implementing the methodology illustrated herein can be used to implement the various aspects of this disclosure. Exemplary hardware that can be used for the disclosed embodiments, configurations and aspects includes computers, handheld devices, telephones (e.g., cellular, Internet enabled, digital, analog, hybrids, and others), and other hardware known in the art. Some of these devices include processors (e.g., a single or multiple microprocessors), memory, nonvolatile storage, input devices, and output devices. Furthermore, alternative software implementations including, but not limited to, distributed processing or component/object distributed processing, parallel processing, or virtual machine processing can also be constructed to implement the methods described herein.

[0166] In yet another embodiment, the disclosed methods may be readily implemented in conjunction with software using object or object-oriented software development environments that provide portable source code that can be used on a variety of computer or workstation platforms. Alternatively, the disclosed system may be implemented partially or fully in hardware using standard logic circuits or VLSI design. Whether software or hardware is used to implement the systems in accordance with this disclosure is dependent on the speed and/or efficiency requirements of the system, the particular function, and the particular software or hardware systems or microprocessor or microcomputer systems being utilized.

[0167] In yet another embodiment, the disclosed methods may be partially implemented in software that can be stored on a storage medium, executed on programmed general-purpose computer with the cooperation of a controller and memory, a special purpose computer, a microprocessor, or the like. In these instances, the systems and methods of this disclosure can be implemented as program embedded on personal computer such as an applet, JAVA® or CGI script, as a resource residing on a server or computer workstation, as a routine embedded in a dedicated measurement system, system component, or the like. The system can also be

implemented by physically incorporating the system and/or method into a software and/or hardware system.

[0168] Although the present disclosure describes components and functions implemented in the aspects, embodiments, and/or configurations with reference to particular standards and protocols, the aspects, embodiments, and/or configurations are not limited to such standards and protocols. Other similar standards and protocols not mentioned herein are in existence and are considered to be included in the present disclosure. Moreover, the standards and protocols mentioned herein and other similar standards and protocols not mentioned herein are periodically superseded by faster or more effective equivalents having essentially the same functions. Such replacement standards and protocols having the same functions are considered equivalents included in the present disclosure.

[0169] The present disclosure, in various aspects, embodiments, and/or configurations, includes components, methods, processes, systems and/or apparatus substantially as depicted and described herein, including various aspects, embodiments, configurations embodiments, subcombinations, and/or subsets thereof. Those of skill in the art will understand how to make and use the disclosed aspects, embodiments, and/or configurations after understanding the present disclosure. The present disclosure, in various aspects, embodiments, and/or configurations, includes providing devices and processes in the absence of items not depicted and/or described herein or in various aspects, embodiments, and/or configurations hereof, including in the absence of such items as may have been used in previous devices or processes, e.g., for improving performance, achieving ease and\or reducing cost of implementation.

[0170] The foregoing discussion has been presented for purposes of illustration and description. The foregoing is not intended to limit the disclosure to the form or forms disclosed herein. In the foregoing Detailed Description for example, various features of the disclosure are grouped together in one or more aspects, embodiments, and/or configurations for the purpose of streamlining the disclosure. The features of the aspects, embodiments, and/or configurations of the disclosure may be combined in alternate aspects, embodiments, and/or configurations other than those discussed above. This method of disclosure is not to be interpreted as reflecting an intention that the claims require more features than are expressly recited in each claim. Rather, as the following claims reflect, inventive aspects lie in less than all features of a single foregoing disclosed aspect, embodiment, and/or configuration. Thus, the following claims are hereby

incorporated into this Detailed Description, with each claim standing on its own as a separate preferred embodiment of the disclosure.

[0171] Moreover, though the description has included description of one or more aspects, embodiments, and/or configurations and certain variations and modifications, other variations, combinations, and modifications are within the scope of the disclosure, e.g., as may be within the skill and knowledge of those in the art, after understanding the present disclosure. It is intended to obtain rights which include alternative aspects, embodiments, and/or configurations to the extent permitted, including alternate, interchangeable and/or equivalent structures, functions, ranges or steps to those claimed, whether or not such alternate, interchangeable and/or equivalent structures, functions, ranges or steps are disclosed herein, and without intending to publicly dedicate any patentable subject matter.

What is claimed is:

1.

2.

ABSTRACT

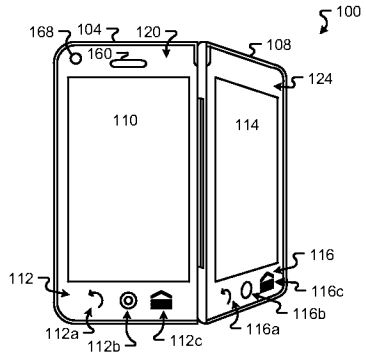


FIG. 1A

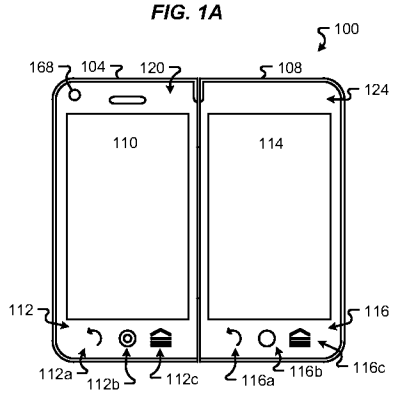


FIG. 1B

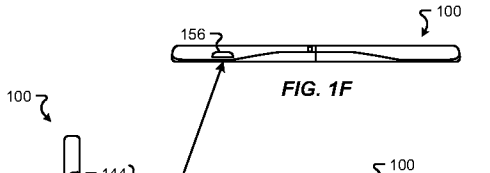


FIG. 1F

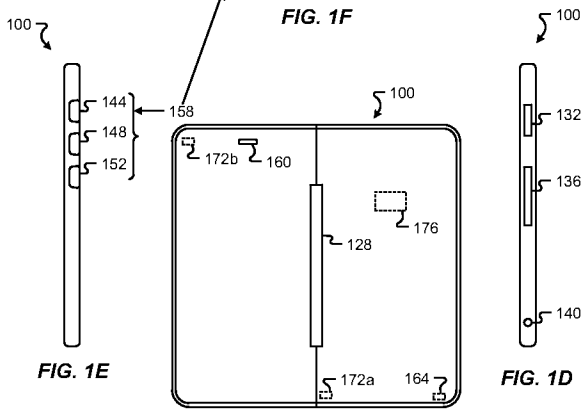


FIG. 1E

FIG. 1C

FIG. 1D

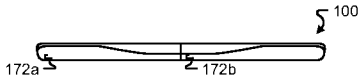


FIG. 1G

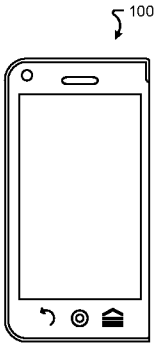


FIG. 1H

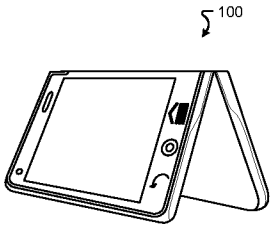


FIG. 1I

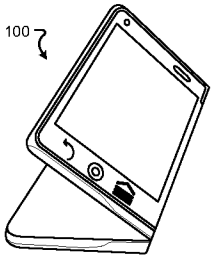


FIG. 1J

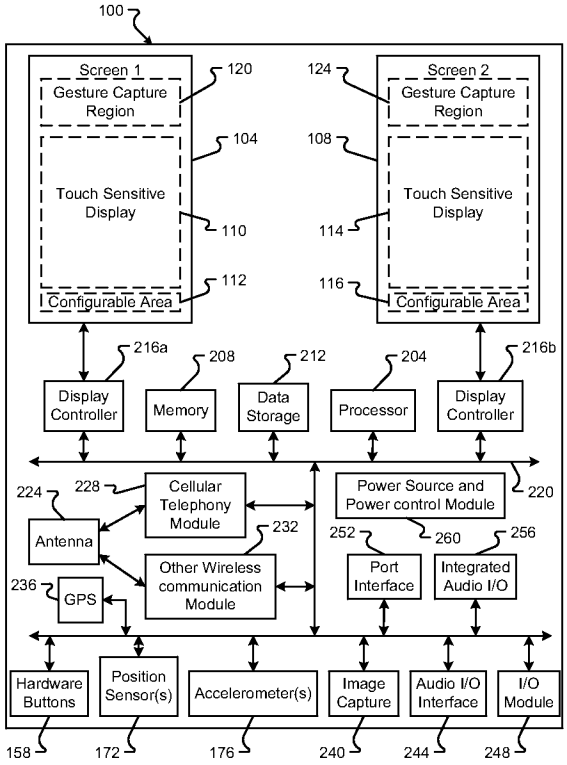


FIG. 2

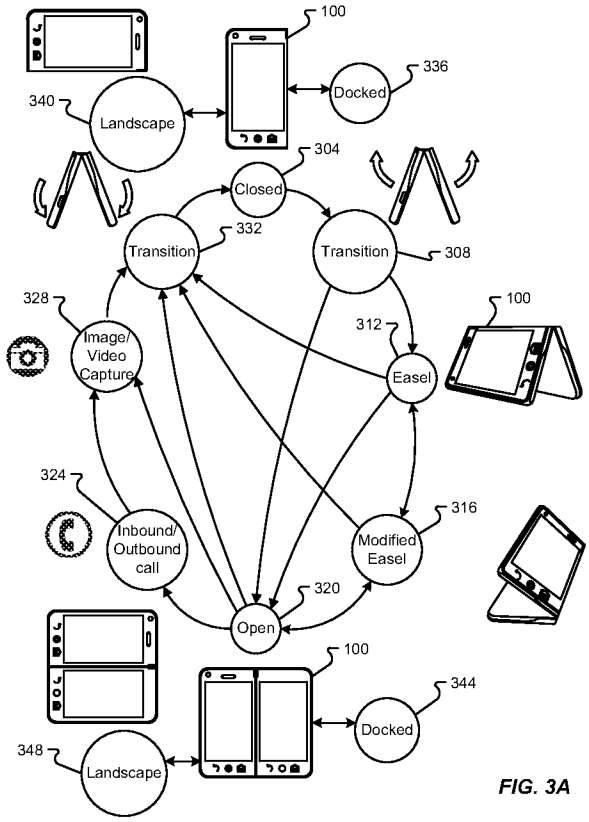


FIG. 3A

		PORTRAIT						LANDSCAPE					
		OPEN	CLOSED	EASEL	MODIFIED EASEL	PHONE	IMAGE / VIDEO	OPEN	CLOSED	EASEL	MODIFIED EASEL	PHONE	IMAGE / VIDEO
PORTRAIT	OPEN	X	HT	HT	HT	P	I	AT	HAT	**HAT**	HAT	P	I
	CLOSED	HT	X	HAT	HAT	P	I	HAT	AT	HAT	HAT	P	I
	EASEL	HT	HT	X	X	P	I	HAT	HAT	HAT	HAT	P	I
	PHONE	HT	X	HT	HT	X	I	HAT	HAT	HAT	HAT	X	I
	IMAGE / VIDEO	HT	HT	HT	X	P	X	HAT	HAT	HAT	HAT	X	HAT
LANDSCAPE	OPEN	AT	HAT	HAT	HAT	P	I	X	HT	HAT	HAT	P	I
	CLOSED	HAT	AT	HAT	HAT	P	I	HT	X	HAT	HAT	P	I
	EASEL	HAT	HAT	HAT	HAT	P	I	HT	HT	X	HAT	P	I
	MODIFIED EASEL	HAT	HAT	HAT	HAT	P	I	HT	HT	HAT	X	P	I
	IMAGE / VIDEO	HAT	HAT	HAT	HAT	HAT	AT	HT	HT	HT	HAT	P	X
								DOCKING SIGNAL					

356 352 372 376 360 368 364

Key:
H - Hall Effect Sensor(s)
a - accelerometer(s)
T- Timer
P – communications Trigger
I – Image / Video capture Request

FIG. 3B

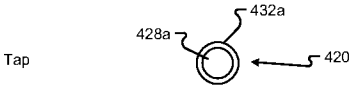


FIG. 4A

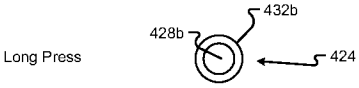


FIG. 4B

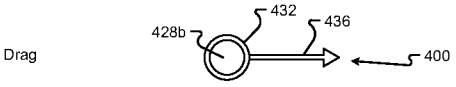


FIG. 4C

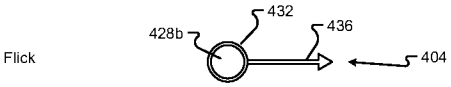


FIG. 4D

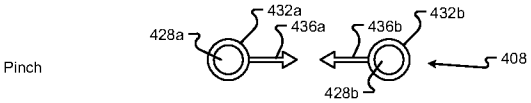


FIG. 4E

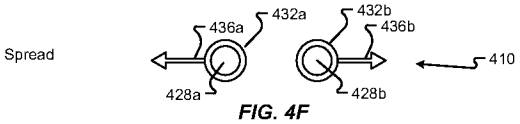


FIG. 4F

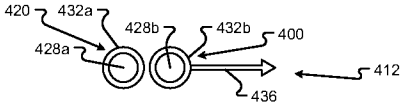


FIG. 4G

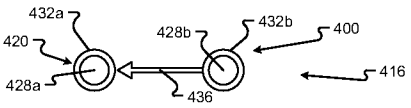


FIG. 4H

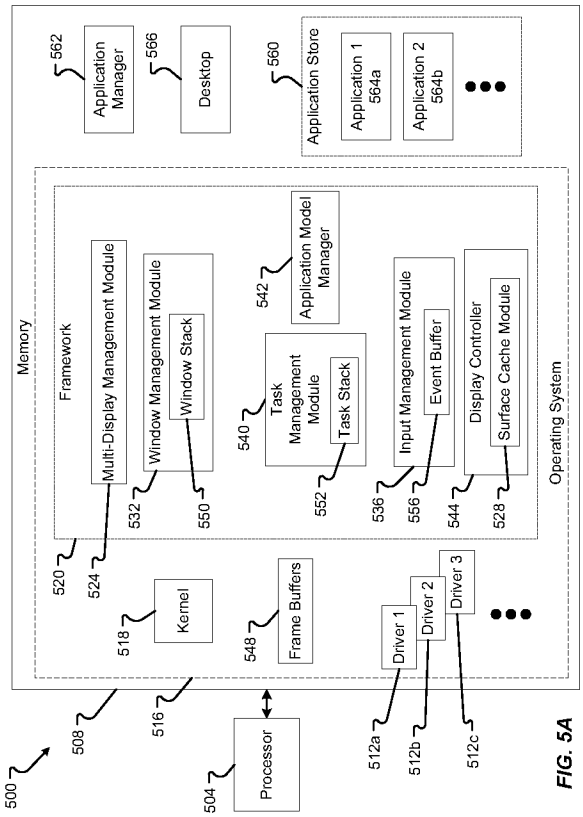


FIG. 5A

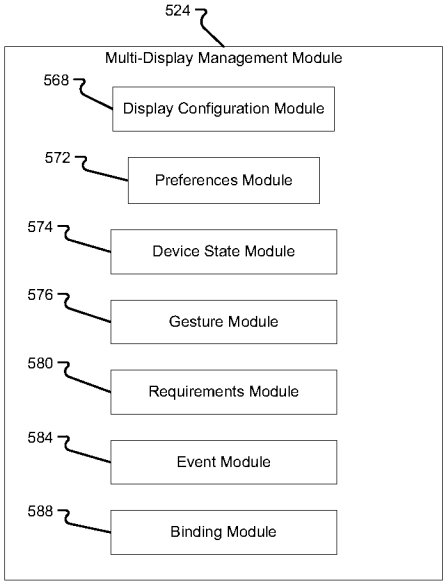


FIG. 5B

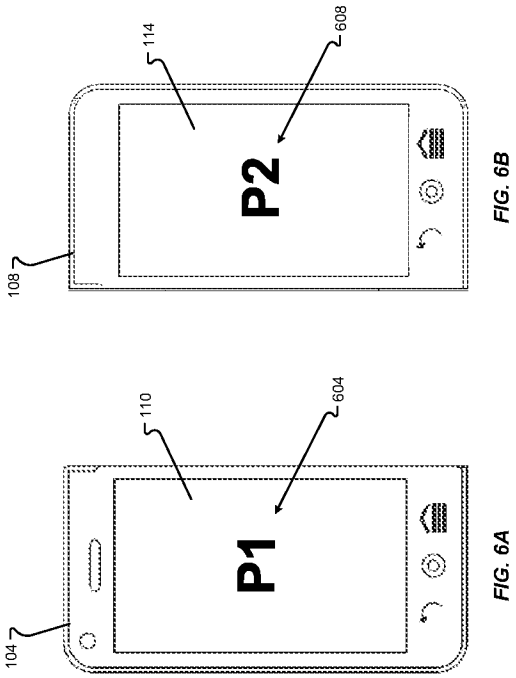


FIG. 6A

FIG. 6B

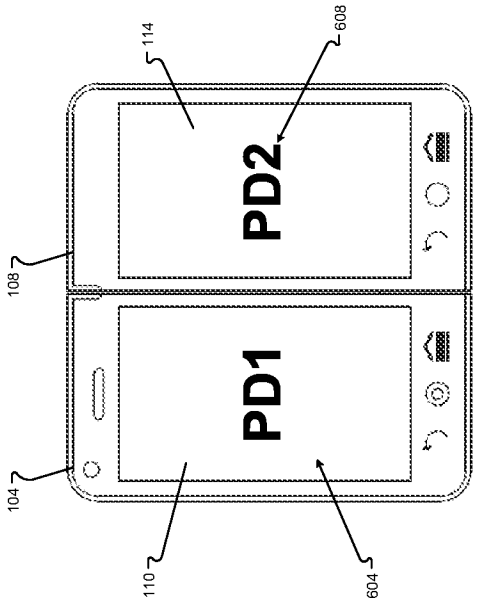


FIG. 6C

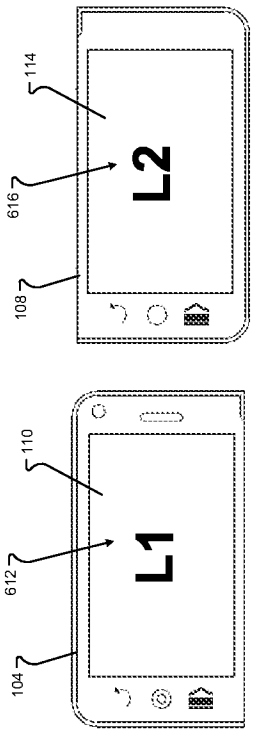


FIG. 6D

FIG. 6E

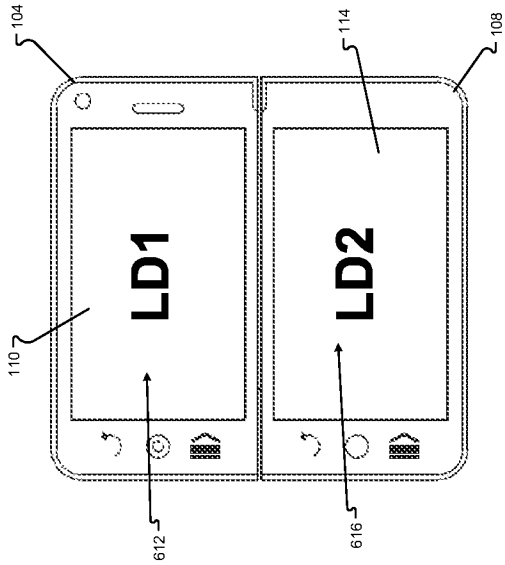


FIG. 6F

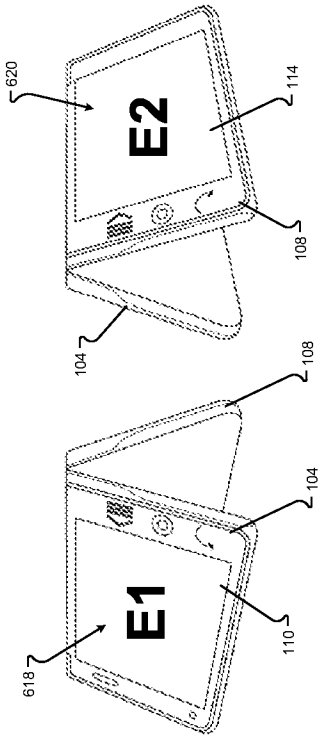


FIG. 6G

FIG. 6H

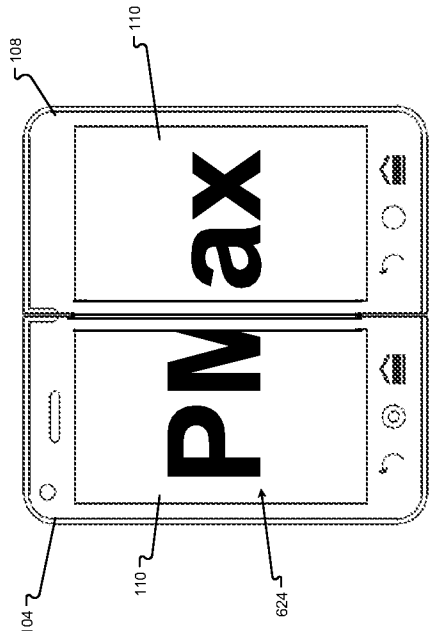


FIG. 6I

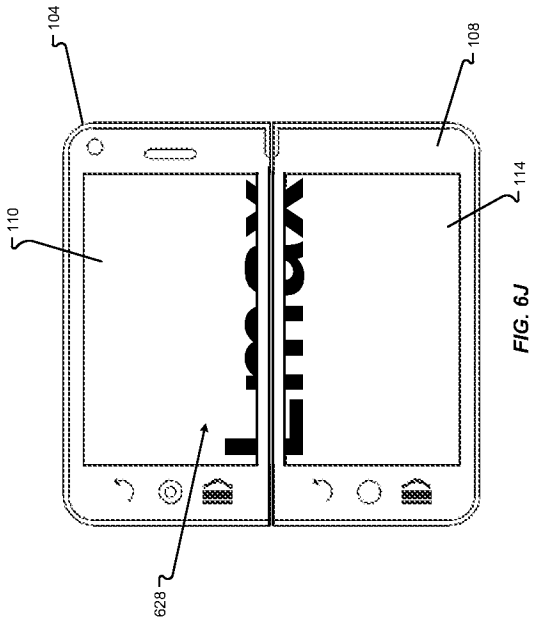


FIG. 6J

[0100] FIGS. 7A-7C illustrate an embodiment showing device 100 and how device 100 responds when a gesture is made. FIG. 7A illustrates device 100 (with screens 104 and 108) when a focus is on screen 104 and display area 110. As noted above, "focus" means active and selected for receiving input. Accordingly, in FIG. 7A screen 104 and display area 110 are active and selected for input. Highlight 704 provides an indication to a user that screen 104 and display area 110 have the focus. If a user presses configurable area 112, device 100 will consider the inputs as inputs for screen 104 and/or for display 110.

[0101] As shown in FIG. 7B, a user can make a gesture 708, such as a tap, drag, or flick gesture on display 114 to indicate a request to launch an application. In some embodiments, the gesture may be made in gesture capture region 120 (e.g., gesture 708A). Device 100 receives the gesture 708 or 708A. In the embodiment shown in FIG. 7, the gesture 708 is made on an icon 716 that represents the application. In response to the gesture 708, an application is launched. As shown in FIG 7C, the launching of the application results in an image 720 being displayed on display 114. The focus is also moved from screen 104 to screen 108, and display area 114, so that image 720 is in focus and a user can immediately begin to interact with the launched application through the image 720. Also, highlighting 704 changes from screen 104 and display area 110 to screen 108 and display area 114. The highlighting 704 provides an indication to a user of what screen, display, and/or application currently has the focus.

[0102] Highlighting 704 as noted above, indicates that screen 104 and display 110 are active. Highlighting 704 can be effected in a number of different ways. In some embodiments, the highlighting 704 is accomplished using light. In these embodiments, a backlight in device 100 may be configured to provide brighter light in some areas, such as around a border as shown in FIGS. 7A-7C. In other embodiments, the brighter light may be located on just one side/area or may be the entire display 110, which is made brighter than at least a portion of screen 108 and/or display 114.

[0103] In other embodiments, highlighting 704 may be accomplished using color, or a displayed image such as an icon that indicates that a screen (104 or 108) and/or a display (110 or 114) have the focus. In other embodiments, the highlighting may be accomplished by some other difference, such as changing text to a different font type, size, or other format change (e.g., italics, underline, bold etc). In some embodiments, the highlighting can be accomplished, or enhanced, by changing a different screen, display, and/or displayed image. For example, in

gesture or flick gesture, but launch applications if gesture 708 is a flick, tap gesture, or double tap gesture. In embodiments, when an application is launched, any image displayed in response to the launching of the application (e.g., open window) is placed in focus. This allows a user to immediately begin to interact with the open application to the displayed image. In some embodiments, the location of the gesture 708, whether in gesture capture region 120 or some other portion of displays 110 and 114, may determine what operation is performed.

[0108] FIGS. 8A-8C illustrates another embodiment of device 100 and its output in response to receiving a gesture. In FIG. 8A the focus is on screen 104, display 110, and image 804. Image 804 is in embodiments a window of an active application which is selected to receive input from a user. In other embodiment, image 804 is an icon on a desktop. In the embodiment shown in FIGS. 8A-8C, a user makes a gesture 708 on display 114. The gesture is a gesture that is associated with launching an application, i.e. a tap on display 114. In response to the gesture 708, device 100 launches an application and then changes focus away from screen 104, display 110, and image 804 and to screen 108 and display 114.

[0109] In some embodiments, as shown in FIG. 8C, the gesture 708 that results in automatically launching an application and placing the focus on an image, i.e., image 808, displayed when the application is launched, does not have to occur on an icon or other representation of an application. In these embodiments, device 100 may be configured by a user so that when the gesture is received on any portion of display 114 (or in other embodiments 110), an application is launched and the focus will be changed on an image, e.g., a window of an open application, displayed as a result of the launched application. The user may, for example, list applications in order of priority such that an application at the top of the list will be launched first. If a window is open for an application at the top the list, then the list can be traversed in order of priority until an application without an open window is found. If all of the applications on the list have open windows then the gesture will not cause any applications to launch, but instead the focus will change to the open window, on display 114 and screen 108, of the application associated with the application at the top of the priority list. In some embodiments, the focus may be changed to a window of the same application that was previously in focus, e.g., a window associated with the same application as image 804. In other embodiments, the gesture 708 may always launch a default application and place the focus on an image displayed as a result of the launch (e.g., open window).

embodiments, the screen, display and/or displayed image that does not have the focus, will be deemphasized by darkening, a color change, or other features that distinguishes it from the highlighted screen, display, and/or image. As can be appreciated, a combination of one or more of the above highlighting features may be used in some embodiments.

[0104] In one embodiment, in addition to highlighting 704, configurable area 112 of screen 104 is highlighted to indicate that screen 104 and display 110 have the focus. Device 100 can be configured so that configurable areas are only highlighted when the corresponding screen has the focus. That is, configurable area 112 is highlighted when screen 104 is in focus, while configurable area 116 is highlighted when focus is on screen 108. As indicated above, the highlighting may be effected by lighting, color, or an image displayed near the configurable area. In the embodiment, shown in FIGS. 7A-7C, when screen 104 is in focus, the configurable area 112 is back lit to show buttons that can be used to receive input from a user.

[0105] In some embodiments, when configurable area 112 is highlighted, input can be received by pressing portions of area 112. At the same time, device 100 may be configured so that no input is received from configurable area 116. Thus, even if a user presses on area 116, no input will be received by device 100. Similarly, when configurable area 116 is highlighted device 100 will allow input to be received from area 116, and no input will be received from area 112. In other embodiments, only one configurable area will be highlighted at a time reflecting the particular screen, display, or image that is in focus. That is, when configurable area 112 is highlighted configurable area 116 will not be highlighted, and when configurable area 116 is highlighted configurable area 112 will not be highlighted.

[0106] It is noted that although gesture 708 is made on display 114, in other embodiments, the gesture 708 may be made on one or more portions of display 110, display 114, configurable area 112, configurable area 116, or as shown in FIG 7B (gesture 708A). The gesture 708 may involve touching portions of screens 104 and/or 108 or can involve only movement above the surface of portions of screens 104 and/or 108. In some embodiments, for the gesture to be recognized, contact is required to be made with the display 110 or 114 for a predetermined period of time.

[0107] As a result of receiving the gesture 708, an application is launched and focus changes from screen 104 and display 110 to screen 108 and display 114. In some embodiments, device 100 is configured to interpret different gestures as indications of different operations. For example, the device 100 may be configured to move an image when the gesture 708 is a drag

[0110] In some embodiments, device 100 is configured so that depending upon the particular gesture 708, an application may be launched and be placed in focus or not. For example, the device 100 may be configured to, in response to gesture 708, launch an application and place the focus on an image displayed as a result of the launch. However, if gesture 708 is a different gesture, a different application may be launched, no application will be launched, or an application may be launched but focus is not changed to an image displayed as a result of the launched application. In other embodiments, the location of the gesture 708, whether in gesture capture region 120 or some other portion of displays 110 and 114, may determine whether an application is launched or not.

[0111] It is noted that although gesture 708 is being made on display 114, in other embodiments, the gesture 708 is made on one or more gesture capture regions (e.g., 120), which are not on display 110 or display 114. The gesture 708 may involve touching some portion of screens 102 or 104 and or can involve only movement above the surface of displays 110 and 114.

[0112] As shown in FIG. 8C, when an application is launched and focus is changed to screen 108 and display 114, configurable area 116 is highlighted to indicate that the focus is changed. Device 100 can be configured so that configurable areas are only highlighted when the corresponding screen has the focus. That is, configurable area 112 is highlighted when screen 104 and/or display 110 is in focus, while configurable area 116 is highlighted when focus is on screen 108 and/or display 114. As indicated above, the highlighting may be effected by lighting, color, or an image displayed near the configurable area. In the embodiment, shown in FIGS. 8A-8C, when screen 104 and/or display 110 is in focus, the configurable area 112 is back lit to show buttons that can be used to receive input from a user. Similarly, when focus changes to screen 108 and/or display 114, the configurable area 116 is back lit to show buttons that can be used to receive input from a user.

[0113] Referring now to FIG. 9, a flow diagram 900 in accordance with at least some embodiments of the present disclosure is shown and will be described. Flow 900 is in embodiments performed by a device such as device 100. More specifically, one or more hardware or software components may be involved in performing flow 900. In one embodiment, modules in middleware 520 (FIG. 5A) such as multi-display management class 524 (FIG. 5B) perform one or more of the steps of flow 900. In other embodiments, in addition to, or in lieu of,

middleware 520 performing steps of flow 900, operating system kernel(s) 516a, processor 504, and/or drivers 512a-512b may also perform steps of flow 900.

[0114] Flow 900 is initiated at 904. Flow 900 passes from 904 to optional step 908 where input regarding focus behavior is received. In this step, a user may provide configuration information that indicates how a device should handle gestures. In embodiments, the device is device 100 (FIGS. 1-3), which includes multiple screens. The user may input any information that affects how a device 100 may handle input gestures, including but not limited to, what applications are launched, changes (if any) that occur to displays 104 and 108 when a gesture has been received, what is highlighted when focus is changed, priority list of applications when a gesture is received, whether focus changes are limited to a display, screen, and/or displayed image, whether the type of gesture received results in different operations, and other information. In one embodiment, the input indicates that in response to a gesture, an application should be launched and focus should be placed on an image displayed as a result of the launching of the application.

[0115] From step 908, flow 900 passes to step 910 where a gesture is received. The gesture can be received in some embodiments by pressure sensitive displays 110 and 114 or gesture capture regions such as regions 120 and 124. In some embodiments, as shown in FIGS. 1 and 2, the gesture capture region(s) is distinct and located away from the pressure sensitive displays. In other embodiments the gesture capture area may overlap partially or completely with the pressure sensitive displays.

[0116] The gesture may be any type of gesture. In embodiments, the gesture is one or more of the gestures illustrated in FIG. 4A-4H. The device 100 is configured so that the gesture received at 910 indicates a request to launch an application and change focus to an image (e.g., window) displayed as a result of launching the application.

[0117] Flow 900 passes from 912 to step 916 where the focus is changed to an image displayed as a result of launching the application. The image may be for example a window that provides a user interface for a user to begin interacting with the application. In some embodiments, this may involve changing the focus from a different window that is displayed on a different display and/or screen. For example, if at the time the gesture is received at step 910 the focus is on a window of a different application, the focus will be changed from the different window to the newly opened window. In other embodiments, there may not be any image that

currently has the focus in which case the focus is placed on the newly opened window. As indicated above, if the newly opened window is on a different screen and display than a window that currently is in focus, the focus will change to the different screen and display where the new the opened window is displayed. The newly opened window, in some embodiments, is displayed on a display and screen which currently displays a window that is in focus. In these instances, the focus will change from a window with the current focus to the newly opened window.

[0118] It should be noted that in some embodiments, step 916 may involve a number of sub steps. For example, in some embodiments as part of step 916, a configurable area may be highlighted, a display may be highlighted, an icon may be generated indicating the focus change, and/or portions of a first screen may be changed to emphasize the highlighting on a second screen. In some embodiments, the substeps may involve not allowing input from a configurable area associated with a first screen and allowing input from a configurable area associated with a second screen.

[0119] The exemplary systems and methods of this disclosure have been described in relation to FIGS 1-10. However, to avoid unnecessarily obscuring the present disclosure, the preceding description omits a number of known structures and devices. This omission is not to be construed as a limitation of the scopes of the claims. Specific details are set forth to provide an understanding of the present disclosure. It should however be appreciated that the present disclosure may be practiced in a variety of ways beyond the specific detail set forth herein.

[0120] Furthermore, while the exemplary aspects, embodiments, and/or configurations illustrated herein show the various components of the system collocated, certain components of the system can be located remotely, at distant portions of a distributed network, such as a LAN and/or the Internet, or within a dedicated system. Thus, it should be appreciated, that the components of the system can be combined in to one or more devices, such as a phone, computer, PDA, electronic book reader, gaming device, or collocated on a particular node of a distributed network, such as an analog and/or digital telecommunications network, a packet-switch network, or a circuit-switched network. It will be appreciated from the preceding description, and for reasons of computational efficiency, that the components of the system can be arranged at any location within a distributed network of components without affecting the operation of the system. For example, the various components can be located in a switch such as a PBX and media server, gateway, in one or more communications devices, at one or more users'

premises, or some combination thereof. Similarly, one or more functional portions of the system could be distributed between a telecommunications device(s) and an associated computing device.

[0121] Furthermore, it should be appreciated that the various links connecting the elements can be wired or wireless links, or any combination thereof, or any other known or later developed element(s) that is capable of supplying and/or communicating data to and from the connected elements. These wired or wireless links can also be secure links and may be capable of communicating encrypted information. Transmission media used as links, for example, can be any suitable carrier for electrical signals, including coaxial cables, copper wire and fiber optics, and may take the form of acoustic or light waves, such as those generated during radio-wave and infra-red data communications.

[0122] Also, while the flowcharts have been discussed and illustrated in relation to a particular sequence of events, it should be appreciated that changes, additions, and omissions to this sequence can occur without materially affecting the operation of the disclosed embodiments, configuration, and aspects.

[0123] A number of variations and modifications of the disclosure can be used. It would be possible to provide for some features of the disclosure without providing others.

[0124] For example in one alternative embodiment, all of the focus behavior information and information regarding launching applications may be preprogrammed into the device 100. In these embodiments, there would be no input received from a user regarding how a device 100 handles gestures indicating the launching of applications or a change in focus.

[0125] In other alternative embodiments, device 100 may be in any configuration described above with respect to FIGS. 6A-6J. In these embodiments, gestures would be received as described above indicating a request for launching an application or changing focus.

[0126] In yet another embodiment, the systems and methods of this disclosure can be implemented in conjunction with a special purpose computer, a programmed microprocessor or microcontroller and peripheral integrated circuit element(s), an ASIC or other integrated circuit, a digital signal processor, a hard-wired electronic or logic circuit such as discrete element circuit, a programmable logic device or gate array such as PLD, PLA, FPGA, PAL, special purpose computer, any comparable means, or the like. In general, any device(s) or means capable of implementing the methodology illustrated herein can be used to implement the various aspects of

this disclosure. Exemplary hardware that can be used for the disclosed embodiments, configurations and aspects includes computers, handheld devices, telephones (e.g., cellular, Internet enabled, digital, analog, hybrids, and others), and other hardware known in the art. Some of these devices include processors (e.g., a single or multiple microprocessors), memory, nonvolatile storage, input devices, and output devices. Furthermore, alternative software implementations including, but not limited to, distributed processing or component/object distributed processing, parallel processing, or virtual machine processing can also be constructed to implement the methods described herein.

[0127] In yet another embodiment, the disclosed methods may be readily implemented in conjunction with software using object or object-oriented software development environments that provide portable source code that can be used on a variety of computer or workstation platforms. Alternatively, the disclosed system may be implemented partially or fully in hardware using standard logic circuits or VLSI design. Whether software or hardware is used to implement the systems in accordance with this disclosure is dependent on the speed and/or efficiency requirements of the system, the particular function, and the particular software or hardware systems or microprocessor or microcomputer systems being utilized.

[0128] In yet another embodiment, the disclosed methods may be partially implemented in software that can be stored on a storage medium, executed on programmed general-purpose computer with the cooperation of a controller and memory, a special purpose computer, a microprocessor, or the like. In these instances, the systems and methods of this disclosure can be implemented as program embedded on personal computer such as an applet, JAVA® or CGI script, as a resource residing on a server or computer workstation, as a routine embedded in a dedicated measurement system, system component, or the like. The system can also be implemented by physically incorporating the system and/or method into a software and/or hardware system.

[0129] Although the present disclosure describes components and functions implemented in the aspects, embodiments, and/or configurations with reference to particular standards and protocols, the aspects, embodiments, and/or configurations are not limited to such standards and protocols. Other similar standards and protocols not mentioned herein are in existence and are considered to be included in the present disclosure. Moreover, the standards and protocols mentioned herein and other similar standards and protocols not mentioned herein are periodically

superseded by faster or more effective equivalents having essentially the same functions. Such replacement standards and protocols having the same functions are considered equivalents included in the present disclosure.

[0130] The present disclosure, in various aspects, embodiments, and/or configurations, includes components, methods, processes, systems and/or apparatus substantially as depicted and described herein, including various aspects, embodiments, configurations embodiments, subcombinations, and/or subsets thereof. Those of skill in the art will understand how to make and use the disclosed aspects, embodiments, and/or configurations after understanding the present disclosure. The present disclosure, in various aspects, embodiments, and/or configurations, includes providing devices and processes in the absence of items not depicted and/or described herein or in various aspects, embodiments, and/or configurations hereof, including in the absence of such items as may have been used in previous devices or processes, e.g., for improving performance, achieving ease and\or reducing cost of implementation.

[0131] The foregoing discussion has been presented for purposes of illustration and description. The foregoing is not intended to limit the disclosure to the form or forms disclosed herein. In the foregoing Detailed Description for example, various features of the disclosure are grouped together in one or more aspects, embodiments, and/or configurations for the purpose of streamlining the disclosure. The features of the aspects, embodiments, and/or configurations of the disclosure may be combined in alternate aspects, embodiments, and/or configurations other than those discussed above. This method of disclosure is not to be interpreted as reflecting an intention that the claims require more features than are expressly recited in each claim. Rather, as the following claims reflect, inventive aspects lie in less than all features of a single foregoing disclosed aspect, embodiment, and/or configuration. Thus, the following claims are hereby incorporated into this Detailed Description, with each claim standing on its own as a separate preferred embodiment of the disclosure.

[0132] Moreover, though the description has included description of one or more aspects, embodiments, and/or configurations and certain variations and modifications, other variations, combinations, and modifications are within the scope of the disclosure, e.g., as may be within the skill and knowledge of those in the art, after understanding the present disclosure. It is intended to obtain rights which include alternative aspects, embodiments, and/or configurations to the extent permitted, including alternate, interchangeable and/or equivalent structures, functions,

ranges or steps to those claimed, whether or not such alternate, interchangeable and/or equivalent structures, functions, ranges or steps are disclosed herein, and without intending to publicly dedicate any patentable subject matter.

What is claimed is:

1. A method, comprising:

displaying a first image on a first display of a first screen, wherein the first image is in focus;

receiving first input indicating a request to launch a second application;

launching the second application;

the second application, displaying a second image on a second display; and

changing the focus from the first image to the second image.

2. The method of claim 1, wherein the first image is displayed in portrait mode on the first display and the second image is displayed in portrait mode on the second display.

3. The method of claim 1, wherein the first image is displayed in landscape mode on the first display and the second image is displayed in landscape mode on the second display.

4. The method of claim 2, further comprising:

wherein the first image is a first open window of a first application and the second image is a second open window of the second application.

5. The method of claim 1, further comprising:

receiving second input indicating that in response to receiving the first input, the second application should be launched and the focus should be placed on the second image displayed by the second application.

6. The method of claim 1, wherein the second display is on a second screen and the method further comprising:

highlighting a configurable area on the first screen when the focus is on the first image in the first display; and

highlighting a second configurable area on the second screen when the focus is on the second image in the second display.

7. The method of claim 1, wherein the second display is on a second screen and further comprising: in response to receiving the first input, highlighting the at least one of the second image, the second display, or the second screen.

8. The method of claim 1, wherein the second display is on a second screen and in response to receiving the input, lighting a second configurable area of a second screen.

9. The method of claim 1, wherein the input is a tap gesture.

10. A non-transitory computer readable medium storing computer executable instructions that when executed by at least one processor perform a method comprising:

receiving input indicating a request to launch an application, the input is received by one of a first touch sensitive display, a second touch sensitive display, or a gesture capture region; and

in response to receiving the input, changing focus from a first image displayed on the first touch sensitive display to a second image displayed by the application.

11. The non-transitory computer readable medium of claim 10, wherein the input is a tap on the second touch sensitive display.

12. The non-transitory computer readable medium of claim 10, wherein the second touch sensitive display is on a second screen and wherein the method further comprises:

in response to receiving the input, lighting a portion of a second configurable area of the second screen and allowing input ftom the second configurable area.

13. The non-transitory computer readable medium of claim 10, wherein the second touch sensitive display is on a second screen and wherein the method further comprises:

in response to receiving the input, darkening a portion of a first configurable area of the first screen and highlighting a portion of a second configurable area of the second screen.

14. The non-transitory computer readable medium of claim 13, wherein the method further comprises:

allowing input from the second configurable area; and

not allowing input from the first configurable area.

15. The non-transitory computer readable medium of claim 10, wherein the method further comprises:

receiving input indicating that in response to the input, the focus should be changed to the second image displayed by the application.

16. The non-transitory computer readable medium of claim 11, wherein the tap requires contact with the second touch sensitive display for a predetermined period of time.

17. A dual screen communication device, comprising:

a first touch sensitive display of a first screen;

a second touch sensitive display of a second screen;

a computer readable medium that stores computer executable instructions that when executed by at least one processor perform a method comprising:

receiving a tap, by the second touch sensitive display of a second screen, indicating a request to launch an application;

launching the application;

the application, displaying a second image on the second touch sensitive display;

changing the focus from the first touch sensitive display of a first screen to the second image; and

allowing input from at least one button of the second screen.

18. The device of claim 17, wherein the method further comprises:

highlighting a portion of the at least one button.

19. The device of claim 17, wherein the method further comprises:

not allowing input from at least a portion of the first touch sensitive display.

20. The device of claim 17, wherein the method further comprises:

highlighting a portion of the second touch sensitive display when the focus is on the second image.

ABSTRACT

Embodiments are described for handling focus when a gesture is input in a multi-screen device. In embodiments, a first image displayed on a first touch sensitive display of a first screen may be currently in focus. In embodiments, the gesture is a tap on a second touch sensitive display of the device. In response to the gesture, an application is launched, which displays a second image on a second display of a second screen. Focus is then changed from the first image on the first touch sensitive display to the second image on the second touch sensitive display.

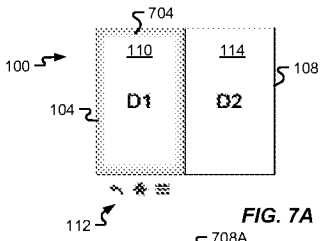


FIG. 7A

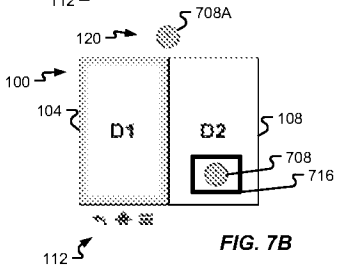


FIG. 7B

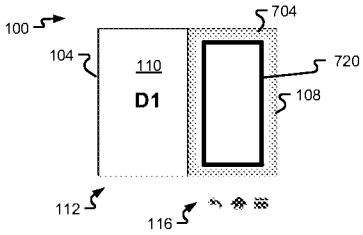


FIG. 7C

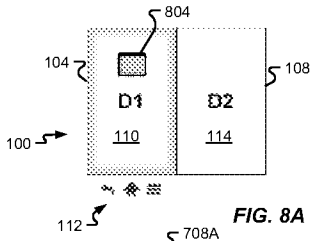


FIG. 8A

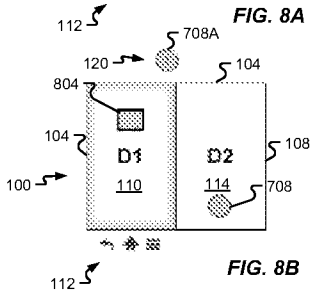


FIG. 8B

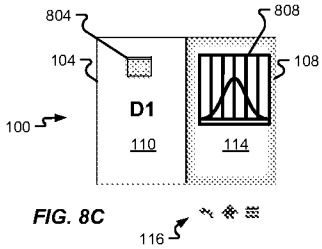


FIG. 8C

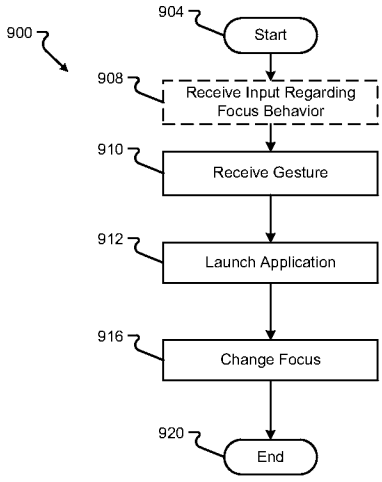


FIG. 9

[0100] FIGS. 7A-7C illustrate an embodiment showing device 100 and how device 100 responds when a gesture is made. FIG. 7A illustrates device 100 (with screens 104 and 108) when a focus is on screen 104 and display 110. As noted above, "focus" means active and selected for receiving input. Accordingly, in FIG. 7A screen 104 and display 110 are active and selected for input. Highlight 704 provides an indication to a user that screen 104 and display 110 is in focus. If a user presses configurable area 112, device 100 will consider the inputs as inputs for screen 104 and/or for display 110.

[0101] As shown in FIG. 7B, a user can make a gesture 708, such as a drag or flick gesture on gesture capture region 120. Device 100 receives the gesture 708. In response to the gesture 708, the focus is placed, e.g. in this embodiment changed from screen 104, on screen 108 and display 110 as shown in FIG. 7C. Also, highlighting 704 changes from screen 104 and display 110 to screen 108 and display 114. The highlighting 704 provides an indication to a user of what screen, display, and/or application currently has the focus.

[0102] Highlighting 704 as noted above, indicates that screen 104 and display 110 are active. Highlighting 704 can be effected in a number of different ways. In some embodiments, the highlighting 704 is accomplished using light. In these embodiments, a backlight in device 100 may be configured to provide brighter light in some areas, such as around a border as shown in FIGS. 7A-7C. In other embodiments, the brighter light may be located on just one side/area or may be the entire display 110, which is made brighter than at least a portion of screen 108 and/or display 114.

[0103] In other embodiments, highlighting 704 may be accomplished using color, or a displayed image such as an icon that indicates that a screen (104 or 108) and/or a display (110 or 114) have the focus. In other embodiments, the highlighting may be accomplished by some other difference, such as changing text to a different font type, size, or other format change (e.g., italics, underline, bold etc). In some embodiments, the highlighting can be accomplished, or enhanced, by changing a different screen, display, and/or displayed image. For example, in embodiments, the screen, display and/or displayed image that does not have the focus, will be deemphasized by darkening, a color change, or other features that distinguishes it from the highlighted screen, display, and/or image. As can be appreciated, a combination of one or more of the above highlighting features may be used in some embodiments.

[0104] In one embodiment, in addition to highlighting 704, configurable area 112 of screen 104 is highlighted to indicate that screen 104 and display 110 have the focus. Device 100 can be configured so that configurable areas are only highlighted when the corresponding screen has the focus. That is, configurable area 112 is highlighted when screen 104 is in focus, while configurable area 116 is highlighted when focus is on screen 108. As indicated above, the highlighting may be effected by lighting, color, or an image displayed near the configurable area. In the embodiment, shown in FIGS. 7A-7C, when screen 104 is in focus, the configurable area 112 is back lit to show buttons that can be used to receive input from a user.

[0105] In some embodiments, when configurable area 112 is highlighted, input can be received by pressing portions of area 112. At the same time, device 100 may be configured so that no input is received from configurable area 116. Thus, even if a user presses on area 116, no input will be received by device 100. Similarly, when configurable area 116 is highlighted device 100 will allow input to be received from area 116, and no input will be received from area 112. In other embodiments, only one configurable area will be highlighted at a time reflecting the particular screen, display, or image that is in focus. That is, when configurable area 112 is highlighted configurable area 116 will not be highlighted, and when configurable area 116 is highlighted configurable area 112 will not be highlighted.

[0106] It is noted that although gesture 708 is being made in gesture capture region 120, which is not on display 110, in other embodiments, the gesture 708 is made on one or more portions of displays 110 and 114. The gesture 708 may involve touching some portion of displays 110 and 114 (which are touch sensitive) or can involve only movement above the surface of displays 110 and 114.

[0107] In some embodiments, device 100 may be configured to make changes to displays 104 and 108 to show that gesture 708 has been received. For example, in some embodiments image 712, which in embodiments is a desktop and may include a number of different image(s) displayed on display 110, appears to move across both displays 110 and 114 when gesture 708 is received. This feature allows a user to appreciate that the gesture 708 has been received, and the focus was changed to screen 108 and display 114 (FIG. 7C). This feature provides some visual feedback to a user.

[0108] FIGS. 8A-8C illustrate another embodiment that also provides feedback to a user about when a gesture is received. FIG. 8A illustrates an unviewed image 804, which cannot be seen by

a user because it is not displayed on any of displays 110 or 114. A user may be aware of the image 804 because of a previous action, such as opening of an application window, moving the image 804 from display 110 or 114. The image 804 may be any type of image such as a photo, a document, an icon that represent an application, or a window of an application. As shown in FIGS. 8A and 8B, the focus is originally on screen 104 and display 110. When gesture 708 is received by gesture capture region 120, image 804 appears to move from an offscreen location to display 114. The movement of image 804 indicates to a user that the gesture 708 was received. As a result of receiving the gesture 708, the focus changes from screen 104 and display 110 to screen 108 and display 114 since image 804 has the focus and is displayed on display 114. This feature allows a user to immediately interact with image 804, since it has the focus. For example, in those embodiments in which image 804 is an open window of an application, a user can begin to interact with the window Afrer making the gesture 704 since the window will have the current focus (see FIG. 8C).

[0109] In some embodiments, device 100 is configured to interpret different gestures as indications of moving different images. For example, the device 100 may be configured to move image 712 (e.g., an entire desktop) when the gesture 708 is a drag gesture and image 804 (active window, icon, photo, document) if gesture 708 is a flick gesture or some other gesture. In other embodiments, the location of the gesture 708, whether in gesture capture region 120 or some other portion of displays 110 and 114, may determine whether image 712 or image 804 is moved.

[0110] FIGS. 9A-9C illustrates another embodiment of device 100 and its output in response to receiving a gesture. In FIG. 9A the focus is on screen 104, display 110, and image 804. Image 804 is in embodiments a window of an active application which is selected to receive input from a user. In the embodiment shown in FIGS. 9A-9C, a user makes a gesture 708 in gesture capture region 120. The gesture is a gesture that is associated with changing focus. In response to the gesture 708, device 100 changes focus away from screen 104, display 110, and image 804 and to screen 108 and display 114.

[0111] In some embodiments, as shown in FIG. 9C, in addition to changing the focus to screen 108 and display 114, the focus is also automatically on an image that is displayed on display 114, i.e., image 904. In these embodiments, device 100 may be configured by a user so that when focus is changed to a screen or display, the focus will also be on an image, e.g., a window of an

open application, if one is open on the display to which the focus has changed. The user may, for example, list applications in order of priority such that if there is an open window the focus will change to an application at the top of the list. If no window is open for an application at the top the list, then the list can be traversed in order of priority until an application with an open window is found. If none of the applications on the list have open windows then the focus will not change to any images on the display. In some embodiments, the focus may be changed to a window of the same application that was previously in focus.

[0112] In other embodiments, the focus may always change to a default image (e.g., open window) selected by a user. In some embodiments, if the default application does not have an open window, device 100 may be configured to launch the application and open a window that will then be in focus.

[0113] In some embodiments, device 100 is configured so that depending upon the particular gesture 708, an image on the newly focused display may be in focus or not. For example, the device 100 may be configured to have in image in focus when the gesture 708 is a drag gesture. If the gesture 708 is a flick gesture or some other gesture, no image will be in focus, and instead only display 114 will be focus. In other embodiments, the location of the gesture 708, whether in gesture capture region 120 or some other portion of displays 110 and 114, may determine whether an image on the newly focused display may be in focus or not.

[0114] It is noted that although gesture 708 is being made in gesture capture region 120, which is not on display 110 or display 114, in other embodiments, the gesture 708 is made on one or more portions of displays 110 and 114. The gesture 708 may involve touching some portion of displays 110 and 114 (which are touch sensitive) or can involve only movement above the surface of displays 110 and 114.

[0115] As shown in FIG. 9C, when focus is changed to screen 108 and display 114, configurable region 116 is highlighted to indicate that the focus is changed. Device 100 can be configured so that configurable regions are only highlighted when the corresponding screen has the focus. That is, configurable area 112 is highlighted when screen 104 and/or display 110 is in focus, while configurable area 116 is highlighted when focus is on screen 108 and/or display 114. As indicated above, the highlighting may be effected by lighting, color, or an image displayed near the configurable area. In the embodiment, shown in FIGS. 9A-9C, when screen 104 and/or display 110 is in focus, the configurable area 112 is back lit to show buttons that can

be used to receive input from a user. Similarly, when focus changes to screen 108 and/or display 114, the configurable area 116 is back lit to show buttons that can be used to receive input from a user.

[0116] Referring now to FIG. 10, a flow diagram 1000 in accordance with at least some embodiments of the present disclosure is shown and will be described. Flow 1000 is in embodiments performed by a device such as device 100. More specifically, one or more hardware or software components may be involved in performing flow 1000. In one embodiment, modules in middleware 520 (FIG. 5A) such as multi-display management class 524 (FIG. 5B) perform one or more of the steps of flow 1000. In other embodiments, in addition to, or in lieu of, middleware 520 performing steps of flow 1000, operating system kernel(s) 516a, processor 504, and/or drivers 512a-512b may also perform steps of flow 1000.

[0117] Flow 1000 is initiated at 1004. Flow 1000 passes from 1004 to optional step 1008 where input regarding focus behavior is received. In this step, a user may provide configuration information that indicates how a device should handle focus when receiving gestures. In embodiments, the device is device 100 (FIGS. 1-3) which includes multiple screens. The user may input any information that affects how a device 100 may handle input gestures, including but not limited to, changes (if any) that occur to displays 104 and 108 when a gesture has been received, what is highlighted when focus is changed, priority list of applications when focus is changed, whether focus changes are limited to a display, screen, and/or displayed image, whether focus behavior changes based on the type of gesture received, and other information. In one embodiment, the input indicates that in response to a gesture, the focus should be placed on the an image that was moved and/or on the screen or display to which the image was moved.

[0118] From step 1008, flow 1000 passes to step 1012 where a gesture is received. The gesture can be received in some embodiments by a gesture capture region such as regions 120 and 124 and/or pressure sensitive displays 110 and 114. In some embodiments, as shown in FIGS. 1 and 2, the gesture capture region(s) is distinct and located away from the pressure sensitive displays. In other embodiments the gesture capture region may overlap partially or completely with the pressure sensitive displays. The gesture is identified as a gesture for moving an image, unviewed or displayed.

[0119] The gesture may be any type of gesture. In embodiments, the gesture is one or more of the gestures illustrated in FIG. 4A-4H. The device 100 is configured so that the gesture received at step1012 indicates a request to move a displayed or an unviewed image.

[0120] In some embodiments, flow 1000 may include an optional step 1016 that changes one or more of the displays to indicate the receipt of the gesture. The change may be for example, showing the movement of an image displayed on a first screen toward a second screen. In other embodiments, the change may comprise showing the movement of an unviewed image to a display where it can be displayed and viewed. The particular changes made to the displays may be based, in part, on the information received at optional step 1008.

[0121] Afrer step 1016, an image is moved at step 1020. The image may be an icon, an open window, a desktop, or a portion of a desktop. As noted above, in embodiments the particular image that is moved will depend upon a number of parameters such as the specific gesture received at step 1012, the location of the gesture, and/or the information received at step 1008 indicating how the gesture should be handled. At step 1020, an image is moved from an original position, which may be off of a display, on a display, or partially on a display, to a new position which may also be off of the display, on a display, or partially on the display.

[0122] Flow 1000 passes from 1020 to step 1024 where the focus is placed on the moved object. In some embodiments, this may involve changing the focus from a different image, display, and/or screen in accordance with predetermined configurations, or maintaining the focus on a display or screen but also placing the focus on a displayed image. For example, if the displayed image that is moved at step 1020 is a window and the original position of the window is on the same screen and display as the new position, then the focus will be maintained on the same screen and display, however the focus will also be placed on the moved image. In other embodiments, if the displayed image that is moved at step 1020 is a window and the original position of the window is on a different screen and a different display than the new position, then the focus will be changed to the different screen and different display, and the focus will also be placed on the moved image. Afrer step 1024, flow 1000 ends at 1028.

[0123] It should be noted that in some embodiments, step 1024 may involve a number of sub steps. For example, in some embodiments as part of step 1024, a configurable area may be highlighted, a display may be highlighted, an icon may be generated indicating the focus change, and/or portions of a first screen may be changed to emphasize the highlighting on a second

screen. In some embodiments, the substeps may involve not allowing input from a configurable area associated with a first screen and allowing input from a configurable area associated with a second screen.

[0124] The exemplary systems and methods of this disclosure have been described in relation to FIGS 1-10. However, to avoid unnecessarily obscuring the present disclosure, the preceding description omits a number of known structures and devices. This omission is not to be construed as a limitation of the scopes of the claims. Specific details are set forth to provide an understanding of the present disclosure. It should however be appreciated that the present disclosure may be practiced in a variety of ways beyond the specific detail set forth herein.

[0125] Furthermore, while the exemplary aspects, embodiments, and/or configurations illustrated herein show the various components of the system collocated, certain components of the system can be located remotely, at distant portions of a distributed network, such as a LAN and/or the Internet, or within a dedicated system. Thus, it should be appreciated, that the components of the system can be combined in to one or more devices, such as a phone, computer, PDA, electronic book reader, gaming device, or collocated on a particular node of a distributed network, such as an analog and/or digital telecommunications network, a packet-switch network, or a circuit-switched network. It will be appreciated from the preceding description, and for reasons of computational efficiency, that the components of the system can be arranged at any location within a distributed network of components without affecting the operation of the system. For example, the various components can be located in a switch such as a PBX and media server, gateway, in one or more communications devices, at one or more users' premises, or some combination thereof. Similarly, one or more functional portions of the system could be distributed between a telecommunications device(s) and an associated computing device.

[0126] Furthermore, it should be appreciated that the various links connecting the elements can be wired or wireless links, or any combination thereof, or any other known or later developed element(s) that is capable of supplying and/or communicating data to and from the connected elements. These wired or wireless links can also be secure links and may be capable of communicating encrypted information. Transmission media used as links, for example, can be any suitable carrier for electrical signals, including coaxial cables, copper wire and fiber optics,

and may take the form of acoustic or light waves, such as those generated during radio-wave and infra-red data communications.

[0127] Also, while the flowcharts have been discussed and illustrated in relation to a particular sequence of events, it should be appreciated that changes, additions, and omissions to this sequence can occur without materially affecting the operation of the disclosed embodiments, configuration, and aspects.

[0128] A number of variations and modifications of the disclosure can be used. It would be possible to provide for some features of the disclosure without providing others.

[0129] For example in one alternative embodiment, all of the focus behavior information may be preprogrammed into the device 100. In these embodiments, there would be no input received by a user regarding how a device 100 handles gestures indicating a request to move an image and how the focus changes.

[0130] In other alternative embodiments, device 100 may be in any configuration described above with respect to FIGS. 6A-6J. In these embodiments, gestures would be received as described above indicating a request for changing focus. The focus would then change from a first screen, display, or image to a second screen, display, or image.

[0131] In yet another embodiment, the systems and methods of this disclosure can be implemented in conjunction with a special purpose computer, a programmed microprocessor or microcontroller and peripheral integrated circuit element(s), an ASIC or other integrated circuit, a digital signal processor, a hard-wired electronic or logic circuit such as discrete element circuit, a programmable logic device or gate array such as PLD, PLA, FPGA, PAL, special purpose computer, any comparable means, or the like. In general, any device(s) or means capable of implementing the methodology illustrated herein can be used to implement the various aspects of this disclosure. Exemplary hardware that can be used for the disclosed embodiments, configurations and aspects includes computers, handheld devices, telephones (e.g., cellular, Internet enabled, digital, analog, hybrids, and others), and other hardware known in the art. Some of these devices include processors (e.g., a single or multiple microprocessors), memory, nonvolatile storage, input devices, and output devices. Furthermore, alternative software implementations including, but not limited to, distributed processing or component/object distributed processing, parallel processing, or virtual machine processing can also be constructed to implement the methods described herein.

[0132] In yet another embodiment, the disclosed methods may be readily implemented in conjunction with software using object or object-oriented software development environments that provide portable source code that can be used on a variety of computer or workstation platforms. Alternatively, the disclosed system may be implemented partially or fully in hardware using standard logic circuits or VLSI design. Whether software or hardware is used to implement the systems in accordance with this disclosure is dependent on the speed and/or efficiency requirements of the system, the particular function, and the particular software or hardware systems or microprocessor or microcomputer systems being utilized.

[0133] In yet another embodiment, the disclosed methods may be partially implemented in software that can be stored on a storage medium, executed on programmed general-purpose computer with the cooperation of a controller and memory, a special purpose computer, a microprocessor, or the like. In these instances, the systems and methods of this disclosure can be implemented as program embedded on personal computer such as an applet, JAVA® or CGI script, as a resource residing on a server or computer workstation, as a routine embedded in a dedicated measurement system, system component, or the like. The system can also be implemented by physically incorporating the system and/or method into a software and/or hardware system.

[0134] Although the present disclosure describes components and functions implemented in the aspects, embodiments, and/or configurations with reference to particular standards and protocols, the aspects, embodiments, and/or configurations are not limited to such standards and protocols. Other similar standards and protocols not mentioned herein are in existence and are considered to be included in the present disclosure. Moreover, the standards and protocols mentioned herein and other similar standards and protocols not mentioned herein are periodically superseded by faster or more effective equivalents having essentially the same functions. Such replacement standards and protocols having the same functions are considered equivalents included in the present disclosure.

[0135] The present disclosure, in various aspects, embodiments, and/or configurations, includes components, methods, processes, systems and/or apparatus substantially as depicted and described herein, including various aspects, embodiments, configurations embodiments, subcombinations, and/or subsets thereof. Those of skill in the art will understand how to make and use the disclosed aspects, embodiments, and/or configurations afrer understanding the

present disclosure. The present disclosure, in various aspects, embodiments, and/or configurations, includes providing devices and processes in the absence of items not depicted and/or described herein or in various aspects, embodiments, and/or configurations hereof, including in the absence of such items as may have been used in previous devices or processes, e.g., for improving performance, achieving ease and\or reducing cost of implementation.

[0136] The foregoing discussion has been presented for purposes of illustration and description. The foregoing is not intended to limit the disclosure to the form or forms disclosed herein. In the foregoing Detailed Description for example, various features of the disclosure are grouped together in one or more aspects, embodiments, and/or configurations for the purpose of streamlining the disclosure. The features of the aspects, embodiments, and/or configurations of the disclosure may be combined in alternate aspects, embodiments, and/or configurations other than those discussed above. This method of disclosure is not to be interpreted as reflecting an intention that the claims require more features than are expressly recited in each claim. Rather, as the following claims reflect, inventive aspects lie in less than all features of a single foregoing disclosed aspect, embodiment, and/or configuration. Thus, the following claims are hereby incorporated into this Detailed Description, with each claim standing on its own as a separate preferred embodiment of the disclosure.

[0137] Moreover, though the description has included description of one or more aspects, embodiments, and/or configurations and certain variations and modifications, other variations, combinations, and modifications are within the scope of the disclosure, e.g., as may be within the skill and knowledge of those in the art, afrer understanding the present disclosure. It is intended to obtain rights which include alternative aspects, embodiments, and/or configurations to the extent permitted, including alternate, interchangeable and/or equivalent structures, functions, ranges or steps to those claimed, whether or not such alternate, interchangeable and/or equivalent structures, functions, ranges or steps are disclosed herein, and without intending to publicly dedicate any patentable subject matter.

What is claimed is:

1. A method, comprising:

receiving, by at least one of a gesture capture region and a touch sensitive display, a gesture indicating a request to move a first unviewed image to one of the touch sensitive display on a first screen or a second touch sensitive display on a second screen;

displaying the first unviewed image in the one of the touch sensitive display or the second touch sensitive display so that it is a first viewed image; and

placing a focus on the viewed image.

2. The method of claim 1, wherein before the displaying, the focus is on a second viewed image in the second touch sensitive display on a second screen and the method further comprises:

changing the focus from the second viewed image to the first viewed image.

3. The method of claim 1, wherein before the displaying, the focus is on a second viewed image in the first touch sensitive display on the first screen and the method further comprises:

changing the focus from the second viewed image to the first viewed image.

4. The method of claim 3, further comprising:

wherein the first displayed image is a displayed window and the second displayed image is a second window.

5. The method of claim 1, further comprising:

receiving input indicating that in response to the gesture, the focus should be changed to the first viewed image.

6. The method of claim 2, further comprising:

highlighting a configurable area on the first screen when the focus is on the first viewed image in the first touch sensitive display; and

highlighting a second configurable area on the second screen when the focus is on the second viewed image in the second touch sensitive display.

7. The method of claim 1, further comprising: in response to receiving the gesture, displaying the unviewed first image as moving from an original position.

8. The method of claim 7, in response to receiving the gesture, displaying the unviewed first image as moved in a direction of the gesture.

9. The method of claim 1, wherein the gesture is a drag gesture.

10. A non-transitory computer readable medium storing computer executable instructions that when executed by at least one processor perform a method comprising:

receiving, by a gesture capture region, a drag gesture indicating a request to move a window;

in response to receiving the drag gesture one or more processors, moving the window from an original position to a new position; and

placing the focus on the window displayed in the new position.

11. The non-transitory computer readable medium of claim 10, wherein the original position is on a display of a first screen and the new position is on a display of a second screen.

12. The non-transitory computer readable medium of claim 11, wherein the method further comprises:

in response to the at least one processor moving the window to the new position, lighting a portion of a second configurable area of the second screen and allowing input from the second configurable area.

13. The non-transitory computer readable medium of claim 11, wherein the method further comprises:

darkening a portion of a first configurable area of the first screen and highlighting a portion of a second configurable area of the second screen.

14. The non-transitory computer readable medium of claim 13, wherein the method further comprises:

allowing input from the second configurable area; and

not allowing input from the first configurable area.

15. The non-transitory computer readable medium of claim 10, wherein the method further comprises:

receiving input indicating that in response to the gesture, the focus should be placed on the window.

16. The non-transitory computer readable medium of claim 10, wherein the original position is off screen and not displayed.

17. A dual screen communication device, comprising:

a gesture capture region to receive a gesture; and

a first touch sensitive display;

a second touch sensitive display;

a computer readable medium that stores computer executable instructions that when executed by at least one processor perform a method comprising:

receiving, by one or more of the gesture capture region, the first touch sensitive display, and the second touch display, a gesture indicating a request to move an image;

moving the image by at least one processor from an original position to a new position in the first touch sensitive display; and

placing a focus on the image afrer the moving.

18. The device of claim 17, wherein the original position is on the second touch sensitive display and the method further comprises:

changing the focus from the second touch sensitive display to the first touch sensitive display.

19. The device of claim 17, wherein the method further comprises:

allowing input from the first touch sensitive display and not allowing input from at least a portion of the second touch sensitive display.

20. The device of claim 17, wherein the method further comprises:

highlighting a portion of the first touch sensitive display when the focus is on the image and the image in on the first touch sensitive display.

ABSTRACT

Embodiments are described for handling focus when a gesture is input in a multi-screen device. In embodiments, the gesture indicates a request to move an image displayed on the multi-screen device. In response, the image is moved and the focus is placed on the moved image.

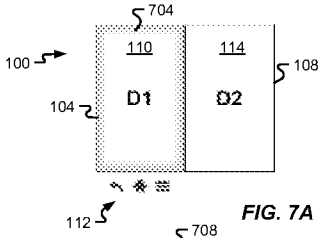


FIG. 7A

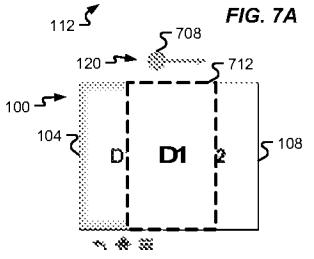


FIG. 7B

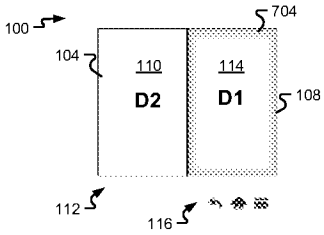


FIG. 7C

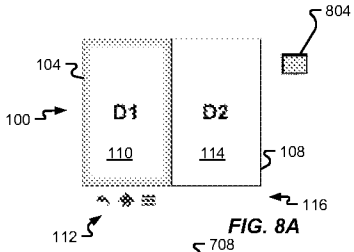


FIG. 8A

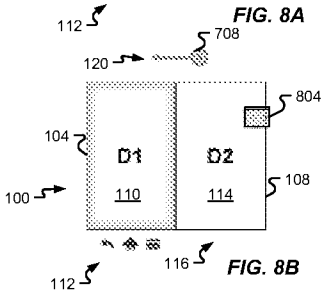


FIG. 8B

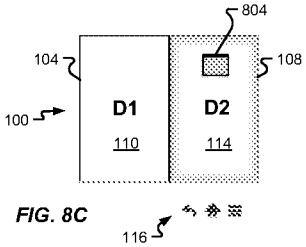


FIG. 8C

116

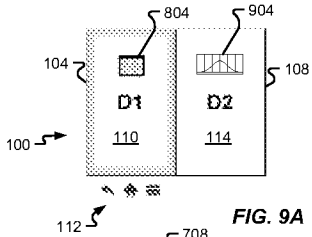


FIG. 9A

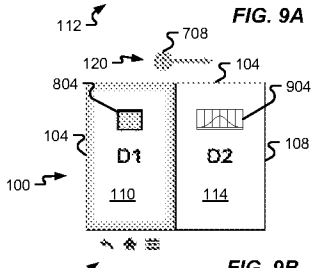


FIG. 9B

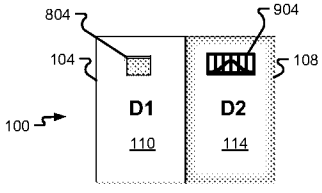


FIG. 9C

116

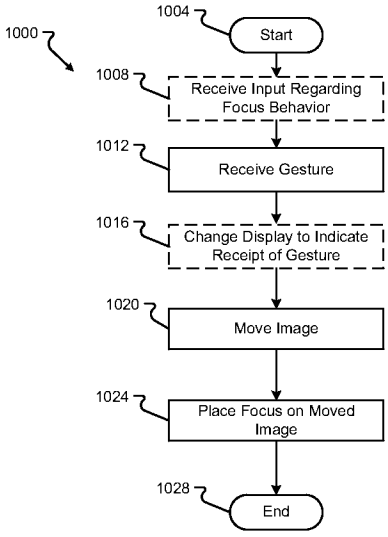


FIG. 10

[0100] FIGS. 7A-7C illustrate an embodiment showing device 100 and how device 100 responds when a gesture is made. FIG. 7A illustrates device 100 (with screens 104 and 108) when a focus is on screen 104 and touch sensitive display 110. As noted above, "focus" means active and selected for receiving input. Accordingly, in FIG. 7A image 106, screen 104, and touch sensitive display 110 are active and selected for input. Highlight 704 provides an indication to a user that image 106, screen 104, and touch sensitive display 110 are in focus. If a user presses configurable area 112, device 100 will consider the inputs as inputs for image 106, screen 104 and/or for display 110.

[0101] As shown in FIG. 7B, a user can make a gesture 708, such as a pinch, drag, or flick gesture on gesture capture region 120. Device 100 receives the gesture 708. In embodiments, the gesture 708 is an indication that the positions of image(s) displayed on touch sensitive display 110 and touch sensitive display 114 should be swapped. In other words, image(s) displayed on touch sensitive display 110 should be displayed on touch sensitive display 114, and image(s) displayed on touch sensitive display 114 should be displayed on touch sensitive display 110.

[0102] As shown in FIG. 7A before the gesture is received, the focus is on image 106 displayed on touch sensitive display 110 of screen 104. In one embodiment, image 106 takes up the entire area of touch sensitive display 110, e.g., an open/maximized window of an application or a desktop, in other embodiments image 106 takes up only a portion of the area of the touch sensitive display, such as an icon, or some portion of a desktop.

[0103] Afrer the gesture 708 is received, the positions of images 106 and 116 (displayed on touch sensitive display 114 of screen 108) are swapped. As shown in FIG. 7C, image 106 is now displayed on touch sensitive display 114 and image 116 is now displayed on touch sensitive display 104. Because prior to receiving the gesture 708, focus was on image 106, after the images 106 and 116 are swapped, the focus will remain on image 106.

[0104] In the embodiment shown in FIGS. 7A-7C, image 106 is displayed in the area of display 114 that previously displayed image 116. Similarly, image 116 is displayed in the area of display 104 that previously displayed image 106. In other embodiments, however, the areas may not completely overlap. That is, image 106 may be displayed on an area of display 114 that only partially overlaps an area where image 116 was displayed. Similarly, image 116 may be

displayed on an area of display 104 that only partially overlaps an area where image 106 was displayed.

[0105] Highlighting 704 as noted above, indicates that image 106 is in focus. Highlighting 704 can be effected in a number of different ways. In some embodiments, the highlighting 704 is accomplished using light. In these embodiments, a backlight in device 100 may be configured to provide brighter light in some areas, such as around a border as shown in FIGS. 7A-7C. In other embodiments, the brighter light may be located on just one side/area or may be the entire display 110, which is made brighter than at least a portion of screen 108 and/or display 114.

[0106] In other embodiments, highlighting 704 may be accomplished using color, or a displayed image such as an icon that indicates that a screen (104 or 108) and/or a display (110 or 114) have the focus. In other embodiments, the highlighting may be accomplished by some other difference, such as changing text to a different font type, size, or other format change (e.g., italics, underline, bold etc). In some embodiments, the highlighting can be accomplished, or enhanced, by changing a different screen, display, and/or displayed image. For example, in embodiments, the screen, display and/or displayed image that does not have the focus, will be deemphasized by darkening, a color change, or other features that distinguishes it from the highlighted screen, display, and/or image. As can be appreciated, a combination of one or more of the above highlighting features may be used in some embodiments.

[0107] In one embodiment, in addition to highlighting 704, configurable area 112 of screen 104 is highlighted to indicate that image 106 displayed on display 110 is in focus. Device 100 can be configured so that configurable areas are only highlighted when a corresponding screen, display and/or an image displayed on the display is in focus. That is, configurable area 112 is highlighted when screen 104, display 110 and/or an image displayed on display 110 is in focus, while configurable area 116 is highlighted when screen 108, display 114 and/or an image displayed on display 114 is in focus. As indicated above, the highlighting 704 may be effected in any number of ways including without limitation lighting, color, design features, and/or an image displayed near the configurable area. In the embodiment, shown in FIGS. 7A-7C, when screen 104 is in focus, the configurable area 112 is back lit to show buttons that can be used to receive input from a user.

[0108] In some embodiments, when configurable area 112 is highlighted, input can be received by pressing portions of area 112. At the same time, device 100 may be configured so that no

input is received from configurable area 116. Thus, even if a user presses on region 116, no input will be received by device 100. Similarly, when configurable area 116 is highlighted device 100 will allow input to be received from configurable area 116, and no input will be received from configurable area 112. In other embodiments, only one configurable area will be highlighted at a time reflecting the particular screen, display, or image that is in focus. That is, when configurable area 112 is highlighted configurable area 116 will not be highlighted, and when configurable area 116 is highlighted configurable area 112 will not be highlighted.

[0109] It is noted that although gesture 708 is being made in gesture capture region 120, which is not on display 110, in other embodiments, the gesture 708 is made on one or more portions of displays 110 and 114. The gesture 708 may involve touching some portion of displays 110 and 114 (which are touch sensitive) or can involve only movement above the surface of displays 110 and 114.

[0110] In some embodiments, device 100 may be configured to make changes to displays 104 and 108 to show that gesture 708 has been received. For example, in some embodiments images 106 and 116, appears to move across both displays 110 and 114 when gesture 708 is received. This feature allows a user to appreciate that the gesture 708 has been received, and the positions of images 106 and 116 are being swapped (FIG. 7C). This feature provides some visual feedback to a user. In the embodiment shown in FIG. 7C, device 100 is configured so that image 106, which is in focus prior to receiving gesture 708, is shown as moving above image 116. In other embodiments, the image that is not in focus may be shown above the other image. In some embodiments, the order (which image is displayed as moving on top) is random. In yet other embodiments, the images may not be shown as moving at all, and instead the images will just be repositioned.

[0111] FIGS. 8A-8C illustrate another embodiment of the present disclosure. FIG. 8A illustrates an image 804 displayed on display 110. The image 804 may be any type of image such as, without limitation, a file, a document, an icon that represent an application, or a window of an application. As shown in FIGS. 8A and 8B, the focus is originally on image 804. In some embodiments, device100 is configured to allow a user to select an image to swap. For example, in embodiments, device 100 will be configured to recognize a first image that is in focus and allow a user to select a second image. The image in focus will then swap positions with the selected image when a predefined gesture is received. This feature may be useful for example in

situations when one desktop (or portion of desktop) on one screen is designated to display icons for frequently used applications and a second desktop (or portion of desktop) on a second screen is used for the remaining applications. The positions of two icons can be quickly swapped from one desktop/screen to the other.

[0112] FIG. 8A shows that image 804 is in focus. FIG. 8B shows that a gesture 708 has been received by the gesture capture region 120. Additionally, FIG. 8B also shows that image 808 has been selected. Image 808 may be selected in any suitable way, such as by tapping on the touch sensitive display 114.

[0113] As shown in FIG. 8B, when gesture 708 is received by gesture capture region 120, image 804 is swapped with image 808. As shown in FIG. 8C after images 804 and 808 have been swapped, image 804 remains in focus. The user can then continue to interact with image 804, since it maintains focus. For example, in those embodiments in which image 804 is an open window of an application, a user can begin to interact with the window afrer making the gesture 704 since the window will have the current focus.

[0114] In some embodiments, device 100 is configured to interpret different gesture locations as indications of moving different images. For example, the device 100 may be configured to swap all images displayed on a display, e.g. an entire desktop or open windows that cover the entire displays, when the gesture 708 is made in gesture capture region 120, however if gesture 708 is made on one of the touch sensitive displays, then an icon or some smaller images on the displays may be swapped. In other embodiments, how the gesture 708 is made may determine whether images that cover entire displays or images that cover only a portion of the displays are swapped. As one example, if a user's fingers are close together when the gesture 708 is made, this may indicate that smaller images should be swapped. In contrast, if a user's fingers are spaced further apart then this may indicate that entire desktop or all of the images displayed on the displays should be swapped.

[0115] Referring now to FIG. 9, a flow diagram 900 in accordance with at least some embodiments of the present disclosure is shown and will be described. Flow 900 is in embodiments performed by a device such as device 100. More specifically, one or more hardware or software components may be involved in performing flow 900. In one embodiment, modules in middleware 520 (FIG. 5A) such as multi-display management class 524 (FIG. 5B) perform one or more of the steps of flow 900. In other embodiments, in addition to, or in lieu of,

middleware 520 performing steps of flow 900, operating system kernel(s) 516a, processor 504, and/or drivers 512a-512b may also perform steps of flow 900.

[0116] Flow 900 is initiated at 104. Flow 900 passes from 104 to optional step 108 where input regarding focus behavior is received. In this step, a user may provide configuration information that indicates how a device should handle focus when receiving gestures. In embodiments, the device is device 100 (FIGS. 1-3) which includes multiple screens. The user may input any information that affects how a device 100 may handle input gestures, including but not limited to, changes (if any) that occur to screens 104 and 108 when a gesture has been received, what is highlighted when focus is changed, priority list of applications when focus is changed, whether focus changes are limited to a display, screen, and/or displayed image, whether focus behavior changes based on the type of gesture received, and other information. In one embodiment, the input indicates that in response to a gesture for swapping images between screens or displays, the focus should be placed on an image that had the focus prior to the images swapping positions.

[0117] Flow 900 passes from step 908 to step 912 where input regarding what images to swap is made. Step 908 is optional because in some embodiments, the images that will be swapped are predetermined. As one example, it is predetermined that an image that is in focus on the display of one screen will be swapped with an image that occupies a large part of the display of a second screen. However, in some embodiments, the display of a first screen may display an image in focus, while the display of a second screen has a number of images, i.e., the desktop with a number of icons. In these situations, step 912 can be performed to allow a user to select an image displayed on the display of the second screen that will be swapped with the image in focus. In other embodiments, step 908 may allow a user to. For example if there is no image in focus a user may be able to select an image on one screen and an image on a second screen to swap. These are merely some examples and it can be appreciated that a user may perform step 912 in some embodiments and not in others.

[0118] From step 912, flow 900 passes to step 916 where a gesture is received. The gesture can be received in some embodiments by a gesture capture region such as regions 120 and 124 and/or pressure sensitive displays 110 and 114. In some embodiments, as shown in FIGS. 1 and 2, the gesture capture region(s) is distinct and located away from the pressure sensitive displays. In other embodiments the gesture capture region may overlap partially or completely with the

pressure sensitive displays 110 and 114. The gesture is identified as a gesture for swapping the position of an image from a first display on a first screen with the position of an image displayed on a second display of a second screen.

[0119] The gesture may be any type of gesture. In embodiments, the gesture is one or a combination of the gestures illustrated in FIG. 4A-4H., such as the pinch gesture. The device 100 is configured so that the gesture received at step 916 indicates a request to move a displayed or an unviewed image.

[0120] In some embodiments, flow 900 may include an optional step 920 that changes one or more of the displays to indicate the receipt of the gesture. The change may be for example, showing the movement of an image displayed on a first screen toward a second screen. As one example, the two images that are swapping positions can be displayed as moving towards their new positions. FIG. 7 illustrates an example of this with image 106 and 116 being shown as moving to their new positions. The particular changes made to the displays may be based, in part, on the information received at optional step 908. That is, a user may select whether there is some change to the displays to indicate that a gesture has been received, and also the particular change, e.g., images move toward their new positions, the image in focus moves above the other image.

[0121] Afrer step 920, the images are moved at step 924. Step 924 involves swapping the position of two images. As indicated above, the images may be, without limitation, an icon, an open window, a desktop, or a portion of a desktop. As noted above, in embodiments the particular image that is moved will depend upon a number of parameters such as the specific gesture received at step 916, the location of the gesture, and/or the information received at step 908 indicating how the gesture should be handled. At step 924, a first image displayed on a display of the first screen is moved from an original position to a new position on a display of a second screen. A second image which is displayed on the display of the second screen is then moved to from its original position to a new position on the display of the first screen.

[0122] Flow 900 passes from 924 to step 928 where the focus is maintained on the moved image that had the focus prior to initiation of flow 900. This allows a user to continue to interact with the image, e.g., an open window of an application, afrer the image has swapped positions. Afrer step 928, flow 900 ends at 932.

[0123] It should be noted that in some embodiments, step 928 may involve a number of sub steps. For example, in some embodiments as part of step 928, a configurable region may be highlighted, a display may be highlighted, an icon may be generated indicating a change in screens or display, such as portions of a first screen may be changed to emphasize the highlighting on a second screen. In some embodiments, the substeps may involve not allowing input from a configurable region associated with a first screen and allowing input from a configurable region associated with a second screen.

[0124] The exemplary systems and methods of this disclosure have been described in relation to FIGS 1-9. However, to avoid unnecessarily obscuring the present disclosure, the preceding description omits a number of known structures and devices. This omission is not to be construed as a limitation of the scopes of the claims. Specific details are set forth to provide an understanding of the present disclosure. It should however be appreciated that the present disclosure may be practiced in a variety of ways beyond the specific detail set forth herein.

[0125] Furthermore, while the exemplary aspects, embodiments, and/or configurations illustrated herein show the various components of the system collocated, certain components of the system can be located remotely, at distant portions of a distributed network, such as a LAN and/or the Internet, or within a dedicated system. Thus, it should be appreciated, that the components of the system can be combined in to one or more devices, such as a phone, computer, PDA, electronic book reader, gaming device, or collocated on a particular node of a distributed network, such as an analog and/or digital telecommunications network, a packet-switch network, or a circuit-switched network. It will be appreciated from the preceding description, and for reasons of computational efficiency, that the components of the system can be arranged at any location within a distributed network of components without affecting the operation of the system. For example, the various components can be located in a switch such as a PBX and media server, gateway, in one or more communications devices, at one or more users' premises, or some combination thereof. Similarly, one or more functional portions of the system could be distributed between a telecommunications device(s) and an associated computing device.

[0126] Furthermore, it should be appreciated that the various links connecting the elements can be wired or wireless links, or any combination thereof, or any other known or later developed element(s) that is capable of supplying and/or communicating data to and from the connected

elements. These wired or wireless links can also be secure links and may be capable of communicating encrypted information. Transmission media used as links, for example, can be any suitable carrier for electrical signals, including coaxial cables, copper wire and fiber optics, and may take the form of acoustic or light waves, such as those generated during radio-wave and infra-red data communications.

[0127] Also, while the flowcharts have been discussed and illustrated in relation to a particular sequence of events, it should be appreciated that changes, additions, and omissions to this sequence can occur without materially affecting the operation of the disclosed embodiments, configuration, and aspects.

[0128] A number of variations and modifications of the disclosure can be used. It would be possible to provide for some features of the disclosure without providing others.

[0129] For example in one alternative embodiment, all of the focus behavior information may be preprogrammed into the device 100. In these embodiments, there would be no input received by a user regarding how a device 100 handles gestures indicating that the position of images should be swapped.

[0130] In other alternative embodiments, device 100 may be in any configuration described above with respect to FIGS. 6A-6J. In these embodiments, gestures would be received as described above indicating a request to swap image positions. The focus would then change with the image originally in focus, meaning the focus would change from a first screen and/or display, to a second screen and/or display.

[0131] In yet another embodiment, the systems and methods of this disclosure can be implemented in conjunction with a special purpose computer, a programmed microprocessor or microcontroller and peripheral integrated circuit element(s), an ASIC or other integrated circuit, a digital signal processor, a hard-wired electronic or logic circuit such as discrete element circuit, a programmable logic device or gate array such as PLD, PLA, FPGA, PAL, special purpose computer, any comparable means, or the like. In general, any device(s) or means capable of implementing the methodology illustrated herein can be used to implement the various aspects of this disclosure. Exemplary hardware that can be used for the disclosed embodiments, configurations and aspects includes computers, handheld devices, telephones (e.g., cellular, Internet enabled, digital, analog, hybrids, and others), and other hardware known in the art. Some of these devices include processors (e.g., a single or multiple microprocessors), memory,

nonvolatile storage, input devices, and output devices. Furthermore, alternative software implementations including, but not limited to, distributed processing or component/object distributed processing, parallel processing, or virtual machine processing can also be constructed to implement the methods described herein.

[0132] In yet another embodiment, the disclosed methods may be readily implemented in conjunction with software using object or object-oriented software development environments that provide portable source code that can be used on a variety of computer or workstation platforms. Alternatively, the disclosed system may be implemented partially or fully in hardware using standard logic circuits or VLSI design. Whether software or hardware is used to implement the systems in accordance with this disclosure is dependent on the speed and/or efficiency requirements of the system, the particular function, and the particular software or hardware systems or microprocessor or microcomputer systems being utilized.

[0133] In yet another embodiment, the disclosed methods may be partially implemented in software that can be stored on a storage medium, executed on programmed general-purpose computer with the cooperation of a controller and memory, a special purpose computer, a microprocessor, or the like. In these instances, the systems and methods of this disclosure can be implemented as program embedded on personal computer such as an applet, JAVA® or CGI script, as a resource residing on a server or computer workstation, as a routine embedded in a dedicated measurement system, system component, or the like. The system can also be implemented by physically incorporating the system and/or method into a software and/or hardware system.

[0134] Although the present disclosure describes components and functions implemented in the aspects, embodiments, and/or configurations with reference to particular standards and protocols, the aspects, embodiments, and/or configurations are not limited to such standards and protocols. Other similar standards and protocols not mentioned herein are in existence and are considered to be included in the present disclosure. Moreover, the standards and protocols mentioned herein and other similar standards and protocols not mentioned herein are periodically superseded by faster or more effective equivalents having essentially the same functions. Such replacement standards and protocols having the same functions are considered equivalents included in the present disclosure.

[0135] The present disclosure, in various aspects, embodiments, and/or configurations, includes components, methods, processes, systems and/or apparatus substantially as depicted and described herein, including various aspects, embodiments, configurations embodiments, subcombinations, and/or subsets thereof. Those of skill in the art will understand how to make and use the disclosed aspects, embodiments, and/or configurations after understanding the present disclosure. The present disclosure, in various aspects, embodiments, and/or configurations, includes providing devices and processes in the absence of items not depicted and/or described herein or in various aspects, embodiments, and/or configurations hereof, including in the absence of such items as may have been used in previous devices or processes, e.g., for improving performance, achieving ease and\or reducing cost of implementation.

[0136] The foregoing discussion has been presented for purposes of illustration and description. The foregoing is not intended to limit the disclosure to the form or forms disclosed herein. In the foregoing Detailed Description for example, various features of the disclosure are grouped together in one or more aspects, embodiments, and/or configurations for the purpose of streamlining the disclosure. The features of the aspects, embodiments, and/or configurations of the disclosure may be combined in alternate aspects, embodiments, and/or configurations other than those discussed above. This method of disclosure is not to be interpreted as reflecting an intention that the claims require more features than are expressly recited in each claim. Rather, as the following claims reflect, inventive aspects lie in less than all features of a single foregoing disclosed aspect, embodiment, and/or configuration. Thus, the following claims are hereby incorporated into this Detailed Description, with each claim standing on its own as a separate preferred embodiment of the disclosure.

[0137] Moreover, though the description has included description of one or more aspects, embodiments, and/or configurations and certain variations and modifications, other variations, combinations, and modifications are within the scope of the disclosure, e.g., as may be within the skill and knowledge of those in the art, after understanding the present disclosure. It is intended to obtain rights which include alternative aspects, embodiments, and/or configurations to the extent permitted, including alternate, interchangeable and/or equivalent structures, functions, ranges or steps to those claimed, whether or not such alternate, interchangeable and/or equivalent structures, functions, ranges or steps are disclosed herein, and without intending to publicly dedicate any patentable subject matter.

What is claimed is:

1. A method, comprising:

displaying by at least one processor a first image on a first display of a first screen;

displaying by the at least one processor a second image on a second display of a second screen, wherein the second image is in focus;

receiving, by at least one of a gesture capture region and a touch sensitive area, a gesture indicating a request to move the first viewed image to the second display and move the second viewed image to the first display;

in response to receiving the gesture:

displaying the first image in the second display;

displaying the second image on the first display; and

maintaining by the at least one processor the focus on the second image in the first display.

2. The method of claim 1, wherein the gesture is a pinch gesture.

3. The method of claim 1, wherein the first image comprises at least a portion of a desktop and the second image comprises at least a second portion of the desktop.

4. The method of claim 1, wherein the first image comprises a first window and the second image comprises a second window.

5. The method of claim 1, wherein the first image comprises a first icon and the second image comprises a second icon.

6. The method of claim 1, further comprising:

receiving input indicating that in response to the gesture, the focus should be maintained on the second image.

7. The method of claim 2, further comprising:

highlighting a second configurable area on the second screen when the second image is displayed on the second display and is in focus; and

highlighting a first configurable area on the first screen when the second image is displayed on the first display and is in focus.

8. The method of claim 1, further comprising:

in response to the at least one processor displaying the second image on the first display, lighting a portion of a first configurable area of the fist screen and allowing input ftom the first configurable area.

9. The method of claim 8, further comprising:

in response to the at least one processor displaying the second image on the first display, darkening a portion of a second configurable area of the second screen.

10. A non-transitory computer readable medium storing computer executable instructions that when executed by at least one processor perform a method comprising:

receiving, by a gesture capture region, a pinch gesture indicating a request to move a first window that is in focus and displayed on a first display of a first screen to a second display of a second screen, and a second window that is displayed on the second display to the first display;

in response to receiving the pinch gesture one or more processors, moving the first window to the second display and the second window to the first display; and

maintaining the focus on the first window on the second display.

11. The non-transitory computer readable medium of claim 10, wherein the moving the first window comprises displaying the first window in an area of the second display that at least partially overlaps an area where the second window was displayed.

12. The non-transitory computer readable medium of claim 10, wherein the moving the second window comprises displaying the second window in an area of the first display that at least partially overlaps an area where the first window was displayed.

13. The non-transitory computer readable medium of claim 11, wherein the method further comprises:

in response to the moving, darkening a portion of a first configurable area of the first screen and highlighting a portion of a second configurable area of the second screen.

14. The non-transitory computer readable medium of claim 11, wherein the method further comprises:

in response to the moving, lighting a portion of a second configurable area of the second screen and allowing input from the second configurable area.

15. The non-transitory computer readable medium of claim 10, wherein the first window is highlighted before receiving the gesture, and the highlighting of the first window is maintained after the moving.

16. The non-transitory computer readable medium of claim 10, wherein the method further comprises:

receiving input indicating that in response to the pinch gesture, the focus should be maintained on a window currently in focus.

17. A dual screen communication device, comprising:

a gesture capture region to receive a gesture; and

a first touch sensitive display;

a second touch sensitive display;

a computer readable medium that stores computer executable instructions that when executed by at least one processor perform a method comprising:

receiving, by one or more of the gesture capture region, the first touch sensitive display, and the second touch sensitive display, a gesture indicating a request to swap positions of a first image displayed on the first touch sensitive display and a second image displayed on the second touch sensitive display;

moving by at least one processor the first image to the second touch sensitive display, and the second image to the first touch sensitive display; and

maintaining a focus on a one of the first image and the second image that has the focus prior to the receiving step.

18. The device of claim 17, wherein the moving comprises displaying the first image in an area of the second display that at least partially overlaps an area where the second window was displayed, and displaying the second image in an area of the first display that at least partially overlaps an area where the first window was displayed.

19. The device of claim 17, wherein the method further comprises:

allowing input from the one of the first touch sensitive display and the second touch sensitive display that is displaying the one of the first image and the second image that has the focus.

20. The device of claim 17, wherein the method further comprises:

not allowing input from the one of the first touch sensitive display and the second touch sensitive display that is not displaying the one of the first image and the second image that has the focus.

ABSTRACT

Embodiments are described for handling focus when a gesture is input in a multi-screen device. In embodiments, the gesture indicates that two images, one that is in focus, swap positions. In response to receiving the gesture, the image in focus is moved from a first display of a first screen to a second display of a second screen. Afrer the images are swapped, the focus is maintained on the image that originally had the focus.

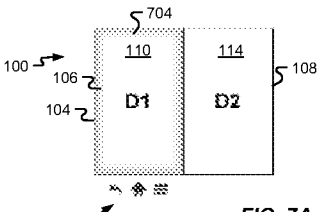


FIG. 7A

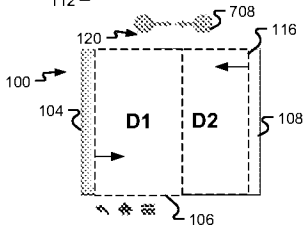


FIG. 7B

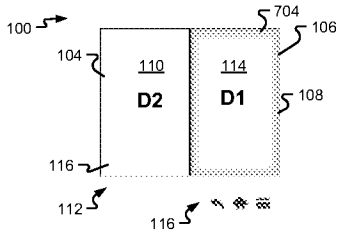


FIG. 7C

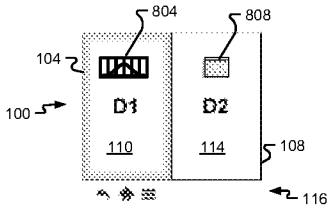


FIG. 8A

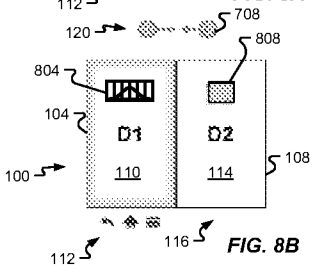


FIG. 8B

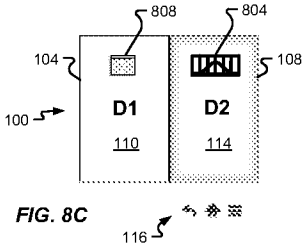


FIG. 8C

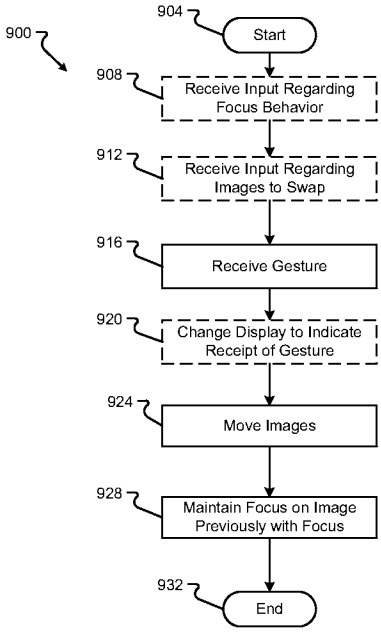


FIG. 10

[0100] FIGS. 7A-7C illustrate an embodiment showing device 100 and how device 100 responds when a gesture is made. FIG. 7A illustrates device 100 (with screens 104 and 108) when a focus is on screen 104 and display area 110. As noted above, "focus" means active and selected for receiving input. Accordingly, in FIG. 7A screen 104 and display area 110 are active and selected for input. Highlight 704 provides an indication to a user that screen 104 and display area 110 have the focus. If a user presses configurable area 112, device 100 will consider the inputs as inputs for screen 104 and/or for display 110.

[0101] As shown in FIG. 7B, a user can make a gesture 708, such as a tap, double tap, drag, or flick gesture on display 114 to indicate a request to change focus to display 114. In other embodiments, the gesture 708A may be made in gesture capture region 120. Device 100 receives the gesture 708 or 708A. In response to the gesture 708 or 708A, the focus is moved from screen 104 to screen 108 and display area 110 as shown in FIG. 7C. Also, highlighting 704 changes from screen 104 and display area 110 to screen 108 and display area 114. The highlighting 704 provides an indication to a user of what screen, display, and/or application currently has the focus.

[0102] Highlighting 704 as noted above, indicates that screen 104 and display 110 are active. Highlighting 704 can be effected in a number of different ways. In some embodiments, the highlighting 704 is accomplished using light. In these embodiments, a backlight in device 100 may be configured to provide brighter light in some areas, such as around a border as shown in FIGS. 7A-7C. In other embodiments, the brighter light may be located on just one side/area or may be the entire display 110, which is made brighter than at least a portion of screen 108 and/or display 114.

[0103] In other embodiments, highlighting 704 may be accomplished using color, or a displayed image such as an icon that indicates that a screen (104 or 108) and/or a display (110 or 114) have the focus. In other embodiments, the highlighting may be accomplished by some other difference, such as changing text to a different font type, size, or other format change (e.g., italics, underline, bold etc). In some embodiments, the highlighting can be accomplished, or enhanced, by changing a different screen, display, and/or displayed image. For example, in embodiments, the screen, display and/or displayed image that does not have the focus, will be deemphasized by darkening, a color change, or other features that distinguishes it from the

highlighted screen, display, and/or image. As can be appreciated, a combination of one or more of the above highlighting features may be used in some embodiments.

[0104] In one embodiment, in addition to highlighting 704, configurable area 112 of screen 104 is highlighted to indicate that screen 104 and display 110 have the focus. Device 100 can be configured so that configurable areas are only highlighted when the corresponding screen has the focus. That is, configurable area 112 is highlighted when screen 104 is in focus, while configurable area 116 is highlighted when focus is on screen 108. As indicated above, the highlighting may be effected by lighting, color, or an image displayed near the configurable area. In the embodiment, shown in FIGS. 7A-7C, when screen 104 is in focus, the configurable area 112 is back lit to show buttons that can be used to receive input from a user.

[0105] In some embodiments, when configurable area 112 is highlighted, input can be received by pressing portions of area 112. At the same time, device 100 may be configured so that no input is received from configurable area 116. Thus, even if a user presses on area 116, no input will be received by device 100. Similarly, when configurable area 116 is highlighted device 100 will allow input to be received from area 116, and no input will be received from area 112. In other embodiments, only one configurable area will be highlighted at a time reflecting the particular screen, display, or image that is in focus. That is, when configurable area 112 is highlighted configurable area 116 will not be highlighted, and when configurable area 116 is highlighted configurable area 112 will not be highlighted.

[0106] It is noted that although gesture 708 is made on display 114, in other embodiments, the gesture 708 may be made on one or more portions of display 110, display 114, configurable area 112, configurable area 116, or as shown in FIG 7B (gesture 708A). The gesture 708 may involve touching portions of screens 104 and/or 108 or can involve only movement above the surface of portions of screens 104 and/or 108.

[0107] As a result of receiving the gesture 708, the focus changes from screen 104 and display 110 to screen 108 and display 114. In some embodiments, device 100 is configured to interpret different gestures as indications of different operations. For example, the device 100 may be configured to move an image when the gesture 708 is a drag gesture or flick gesture, but change focus if gesture 708 is a flick, tap gesture, or double tap gesture. In other embodiments, the location of the gesture 708, whether in gesture capture region 120 or some other portion of displays 110 and 114, may determine what operation is performed.

[0108] FIGS. 8A-8C illustrates another embodiment of device 100 and its output in response to receiving a gesture. In FIG. 8A the focus is on screen 104, display 110, and image 804. Image 804 is in embodiments a window of an active application which is selected to receive input from a user. In other embodiment, image 804 is an icon on a desktop. In the embodiment shown in FIGS. 8A-8C, a user makes a gesture 708 on display 114. The gesture is a gesture that is associated with changing focus, i.e. a tap on display 114. In response to the gesture 708, device 100 changes focus away from screen 104, display 110, and image 804 and to screen 108 and display 114. In some embodiments, for the gesture to be recognized, is required to be made with the display 110 or 114 for a predetermined period of time.

[0109] In some embodiments, as shown in FIG. 8C, in addition to changing the focus to screen 108 and display 114, the focus is also automatically placed on an image that is displayed on display 114, i.e., image 808. In these embodiments, device 100 may be configured by a user so that when focus is changed to a screen or display, the focus will also be on an image, e.g., a window of an open application, if one is open on the display to which the focus has changed. The user may, for example, list applications in order of priority such that if there is not an open window the focus will change to an application at the top of the list. If no window is open for an application at the top the list, then the list can be traversed in order of priority until an application with an open window is found. If none of the applications on the list have open windows then the focus will not change to any images on the display and display 114 and screen 108 will be active. In some embodiments, the focus may be changed to a window of the same application that was previously in focus.

[0110] In other embodiments, the focus may always change to a default image (e.g., open window) selected by a user. In some embodiments, if the default application does not have an open window, device 100 may be configured to launch the application and open a window that will then be in focus.

[0111] In some embodiments, device 100 is configured so that depending upon the particular gesture 708, an image on the newly focused display may be in focus or not. For example, the device 100 may be configured to have in image in focus when the gesture 708 is a tap gesture on one of the displays 110 or 114. However, if gesture 708 is a drag gesture, flick, or some other gesture, no image will be in focus, and instead only display 114 will be focus. In other embodiments, the location of the gesture 708, whether in gesture capture region 120 or some

other portion of displays 110 and 114, may determine whether an image on the newly focused display may be in focus or not.

[0112] It is noted that although gesture 708 is being made on display 114, in other embodiments, the gesture 708 is made on one or more gesture capture regions (e.g., 120), which are not on display 110 or display 114. The gesture 708 may involve touching some portion of screens 102 or 104 and or can involve only movement above the surface of displays 110 and 114.

[0113] As shown in FIG. 8C, when focus is changed to screen 108 and display 114, configurable area 116 is highlighted to indicate that the focus is changed. Device 100 can be configured so that configurable areas are only highlighted when the corresponding screen has the focus. That is, configurable area 112 is highlighted when screen 104 and/or display 110 is in focus, while configurable area 116 is highlighted when focus is on screen 108 and/or display 114. As indicated above, the highlighting may be effected by lighting, color, or an image displayed near the configurable area. In the embodiment, shown in FIGS. 8A-8C, when screen 104 and/or display 110 is in focus, the configurable area 112 is back lit to show buttons that can be used to receive input from a user. Similarly, when focus changes to screen 108 and/or display 114, the configurable area 116 is back lit to show buttons that can be used to receive input from a user.

[0114] Referring now to FIG. 9, a flow diagram 900 in accordance with at least some embodiments of the present disclosure is shown and will be described. Flow 900 is in embodiments performed by a device such as device 100. More specifically, one or more hardware or software components may be involved in performing flow 900. In one embodiment, modules in middleware 520 (FIG. 5A) such as multi-display management class 524 (FIG. 5B) perform one or more of the steps of flow 900. In other embodiments, in addition to, or in lieu of, middleware 520 performing steps of flow 900, operating system kernel(s) 516a, processor 504, and/or drivers 512a-512b may also perform steps of flow 900.

[0115] Flow 900 is initiated at 904. Flow 900 passes from 904 to optional step 908 where input regarding focus behavior is received. In this step, a user may provide configuration information that indicates how a device should handle focus when receiving gestures. In embodiments, the device is device 100 (FIGS. 1-3), which includes multiple screens. The user may input any information that affects how a device 100 may handle input gestures, including but not limited to, changes (if any) that occur to displays 104 and 108 when a gesture has been

received, what is highlighted when focus is changed, priority list of applications when focus is changed, whether focus changes are limited to a display, screen, and/or displayed image, whether focus behavior changes based on the type of gesture received, and other information. In one embodiment, the input indicates that in response to a gesture, the focus should be placed on the display that received the gesture, such as the tap, drag, or flick.

[0116] From step 908, flow 900 passes to step 912 where a gesture is received. The gesture can be received in some embodiments by pressure sensitive displays 110 and 114 or gesture capture regions such as regions 120 and 124. In some embodiments, as shown in FIGS. 1 and 2, the gesture capture region(s) is distinct and located away from the pressure sensitive displays. In other embodiments the gesture capture area may overlap partially or completely with the pressure sensitive displays. The gesture is identified as a gesture for moving an image, unviewed or displayed.

[0117] The gesture may be any type of gesture. In embodiments, the gesture is one or more of the gestures illustrated in FIG. 4A-4H. The device 100 is configured so that the gesture received at 912 indicates a request to change focus.

[0118] Flow 900 passes from 912 to step 916 where the focus is changed. In some embodiments, this may involve changing the focus from a different image, display, and/or screen in accordance with predetermined configurations. For example, if the gesture indicates to move the focus to a different screen and display (e.g., a tap on the different display), the focus will be changed to the different screen or display. In other embodiments, the gesture may specifically indicate changing focus to a displayed image (e.g., tap on an image), in which case the focus will be changed to the specific image. In some embodiments, the tapped image may be in the same screen or display as the image currently in focus. The tap would then result in simply changing the focus to the tapped image. As indicated above, if the image is on a different screen and display, the focus changes to the different screen and display.

[0119] It should be noted that in some embodiments, step 916 may involve a number of sub steps. For example, in some embodiments as part of step 916, a configurable area may be highlighted, a display may be highlighted, an icon may be generated indicating the focus change, and/or portions of a first screen may be changed to emphasize the highlighting on a second screen. In some embodiments, the substeps may involve not allowing input from a configurable

area associated with a first screen and allowing input from a configurable area associated with a second screen.

[0120] The exemplary systems and methods of this disclosure have been described in relation to FIGS 1-10. However, to avoid unnecessarily obscuring the present disclosure, the preceding description omits a number of known structures and devices. This omission is not to be construed as a limitation of the scopes of the claims. Specific details are set forth to provide an understanding of the present disclosure. It should however be appreciated that the present disclosure may be practiced in a variety of ways beyond the specific detail set forth herein.

[0121] Furthermore, while the exemplary aspects, embodiments, and/or configurations illustrated herein show the various components of the system collocated, certain components of the system can be located remotely, at distant portions of a distributed network, such as a LAN and/or the Internet, or within a dedicated system. Thus, it should be appreciated, that the components of the system can be combined in to one or more devices, such as a phone, computer, PDA, electronic book reader, gaming device, or collocated on a particular node of a distributed network, such as an analog and/or digital telecommunications network, a packet-switch network, or a circuit-switched network. It will be appreciated from the preceding description, and for reasons of computational efficiency, that the components of the system can be arranged at any location within a distributed network of components without affecting the operation of the system. For example, the various components can be located in a switch such as a PBX and media server, gateway, in one or more communications devices, at one or more users' premises, or some combination thereof. Similarly, one or more functional portions of the system could be distributed between a telecommunications device(s) and an associated computing device.

[0122] Furthermore, it should be appreciated that the various links connecting the elements can be wired or wireless links, or any combination thereof, or any other known or later developed element(s) that is capable of supplying and/or communicating data to and from the connected elements. These wired or wireless links can also be secure links and may be capable of communicating encrypted information. Transmission media used as links, for example, can be any suitable carrier for electrical signals, including coaxial cables, copper wire and fiber optics, and may take the form of acoustic or light waves, such as those generated during radio-wave and infra-red data communications.

[0123] Also, while the flowcharts have been discussed and illustrated in relation to a particular sequence of events, it should be appreciated that changes, additions, and omissions to this sequence can occur without materially affecting the operation of the disclosed embodiments, configuration, and aspects.

[0124] A number of variations and modifications of the disclosure can be used. It would be possible to provide for some features of the disclosure without providing others.

[0125] For example in one alternative embodiment, all of the focus behavior information may be preprogrammed into the device 100. In these embodiments, there would be no input received from a user regarding how a device 100 handles gestures indicating a change in focus.

[0126] In other alternative embodiments, device 100 may be in any configuration described above with respect to FIGS. 6A-6J. In these embodiments, gestures would be received as described above indicating a request for changing focus. The focus would then change from a first screen, display, or image to a second screen, display, or image.

[0127] In yet another embodiment, the systems and methods of this disclosure can be implemented in conjunction with a special purpose computer, a programmed microprocessor or microcontroller and peripheral integrated circuit element(s), an ASIC or other integrated circuit, a digital signal processor, a hard-wired electronic or logic circuit such as discrete element circuit, a programmable logic device or gate array such as PLD, PLA, FPGA, PAL, special purpose computer, any comparable means, or the like. In general, any device(s) or means capable of implementing the methodology illustrated herein can be used to implement the various aspects of this disclosure. Exemplary hardware that can be used for the disclosed embodiments, configurations and aspects includes computers, handheld devices, telephones (e.g., cellular, Internet enabled, digital, analog, hybrids, and others), and other hardware known in the art. Some of these devices include processors (e.g., a single or multiple microprocessors), memory, nonvolatile storage, input devices, and output devices. Furthermore, alternative software implementations including, but not limited to, distributed processing or component/object distributed processing, parallel processing, or virtual machine processing can also be constructed to implement the methods described herein.

[0128] In yet another embodiment, the disclosed methods may be readily implemented in conjunction with software using object or object-oriented software development environments that provide portable source code that can be used on a variety of computer or workstation

platforms. Alternatively, the disclosed system may be implemented partially or fully in hardware using standard logic circuits or VLSI design. Whether software or hardware is used to implement the systems in accordance with this disclosure is dependent on the speed and/or efficiency requirements of the system, the particular function, and the particular software or hardware systems or microprocessor or microcomputer systems being utilized.

[0129] In yet another embodiment, the disclosed methods may be partially implemented in software that can be stored on a storage medium, executed on programmed general-purpose computer with the cooperation of a controller and memory, a special purpose computer, a microprocessor, or the like. In these instances, the systems and methods of this disclosure can be implemented as program embedded on personal computer such as an applet, JAVA® or CGI script, as a resource residing on a server or computer workstation, as a routine embedded in a dedicated measurement system, system component, or the like. The system can also be implemented by physically incorporating the system and/or method into a software and/or hardware system.

[0130] Although the present disclosure describes components and functions implemented in the aspects, embodiments, and/or configurations with reference to particular standards and protocols, the aspects, embodiments, and/or configurations are not limited to such standards and protocols. Other similar standards and protocols not mentioned herein are in existence and are considered to be included in the present disclosure. Moreover, the standards and protocols mentioned herein and other similar standards and protocols not mentioned herein are periodically superseded by faster or more effective equivalents having essentially the same functions. Such replacement standards and protocols having the same functions are considered equivalents included in the present disclosure.

[0131] The present disclosure, in various aspects, embodiments, and/or configurations, includes components, methods, processes, systems and/or apparatus substantially as depicted and described herein, including various aspects, embodiments, configurations embodiments, subcombinations, and/or subsets thereof. Those of skill in the art will understand how to make and use the disclosed aspects, embodiments, and/or configurations afrer understanding the present disclosure. The present disclosure, in various aspects, embodiments, and/or configurations, includes providing devices and processes in the absence of items not depicted and/or described herein or in various aspects, embodiments, and/or configurations hereof,

including in the absence of such items as may have been used in previous devices or processes, e.g., for improving performance, achieving ease and\or reducing cost of implementation.

[0132] The foregoing discussion has been presented for purposes of illustration and description. The foregoing is not intended to limit the disclosure to the form or forms disclosed herein. In the foregoing Detailed Description for example, various features of the disclosure are grouped together in one or more aspects, embodiments, and/or configurations for the purpose of streamlining the disclosure. The features of the aspects, embodiments, and/or configurations of the disclosure may be combined in alternate aspects, embodiments, and/or configurations other than those discussed above. This method of disclosure is not to be interpreted as reflecting an intention that the claims require more features than are expressly recited in each claim. Rather, as the following claims reflect, inventive aspects lie in less than all features of a single foregoing disclosed aspect, embodiment, and/or configuration. Thus, the following claims are hereby incorporated into this Detailed Description, with each claim standing on its own as a separate preferred embodiment of the disclosure.

[0133] Moreover, though the description has included description of one or more aspects, embodiments, and/or configurations and certain variations and modifications, other variations, combinations, and modifications are within the scope of the disclosure, e.g., as may be within the skill and knowledge of those in the art, after understanding the present disclosure. It is intended to obtain rights which include alternative aspects, embodiments, and/or configurations to the extent permitted, including alternate, interchangeable and/or equivalent structures, functions, ranges or steps to those claimed, whether or not such alternate, interchangeable and/or equivalent structures, functions, ranges or steps are disclosed herein, and without intending to publicly dedicate any patentable subject matter.

What is claimed is:

1. A method, comprising:

displaying a first image in a first display of a first screen, wherein the first image is in focus;

displaying a second image in a second display of a second screen;

receiving, by at least one of a gesture capture region and a touch sensitive display, a gesture indicating a request to change focus from at least one of the first image to the second image, the second display, or the second screen; and

changing the focus from the first image to at least one of the second image, the second display, or the second screen.

2. The method of claim 1, wherein the changing comprises:

changing the focus from the first image to the second image.

3. The method of claim 1, wherein the changing comprises:

changing the focus from the first image to the second display.

4. The method of claim 2, further comprising:

wherein the first displayed image is a first open window of a first application and the second displayed image is a second open window of a second application.

5. The method of claim 1, further comprising:

receiving input indicating that in response to the gesture, the focus should be changed to the at least one of the second image, the second display, or the second screen.

6. The method of claim 1, further comprising:

highlighting a configurable area on the first screen when the focus is on the first image in the first display area; and

highlighting a second configurable area on the second screen when the focus is on the at least one of the second image, the second display, or the second screen.

7. The method of claim 1, further comprising: in response to receiving the gesture, highlighting the at least one of the second image, the second display, or the second screen.

8. The method of claim 1, in response to receiving the gesture, lighting a second configurable area of the second screen.

9. The method of claim 1, wherein the gesture is a tap on the second touch sensitive display.

10. A non-transitory computer readable medium storing computer executable instructions that when executed by at least one processor perform a method comprising:

receiving, by a second touch sensitive display of a second screen, input indicating a request to change focus to the second touch sensitive display; and

in response to receiving the input, changing focus from a first image displayed on a first touch sensitive display of a first screen to the second touch sensitive display.

11. The non-transitory computer readable medium of claim 10, wherein the input is a tap on the second touch sensitive display.

12. The non-transitory computer readable medium of claim 10, wherein the method further comprises:

in response to receiving the input, lighting a portion of a second configurable area of the second screen and allowing input from the second configurable area.

13. The non-transitory computer readable medium of claim 10, wherein the method further comprises:

in response to receiving the input, darkening a portion of a first configurable area of the first screen and highlighting a portion of a second configurable area of the second screen.

14. The non-transitory computer readable medium of claim 13, wherein the method further comprises:

allowing input from the second configurable area; and

not allowing input from the first configurable area.

15. The non-transitory computer readable medium of claim 10, wherein the method further comprises:

receiving input indicating that in response to the input, the focus should be changed to the second touch sensitive display.

16. The non-transitory computer readable medium of claim 11, wherein the tap requires contact with the screen for a predetermined period of time.

17. A dual screen communication device, comprising:

a first touch sensitive display of a first screen;

a second touch sensitive display of a second screen;

a computer readable medium that stores computer executable instructions that when executed by at least one processor perform a method comprising:

receiving a tap, by the second touch display, indicating a request to change focus;

changing the focus from the first touch sensitive display to the second touch sensitive display; and

allowing input from at least one button of the second screen.

18. The device of claim 17, wherein the method further comprises:

highlighting a portion of the at least one button.

19. The device of claim 17, wherein the method further comprises:

not allowing input from at least a portion of the first touch sensitive display.

20. The device of claim 17, wherein the method further comprises:

highlighting a portion of the second touch sensitive display when the focus is on the second touch sensitive display.

ABSTRACT

Embodiments are described for handling focus when a gesture is input in a multi-screen device. In embodiments, a first image displayed on a first touch sensitive display of a first screen may be currently in focus. In embodiments, the gesture is a tap on a second touch sensitive display of the device. In response to the gesture, focus is changed from the first image on the first touch sensitive display to the second touch sensitive display.

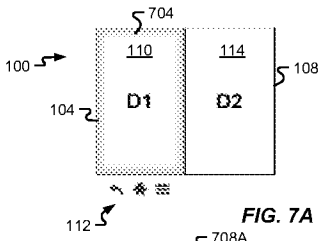


FIG. 7A

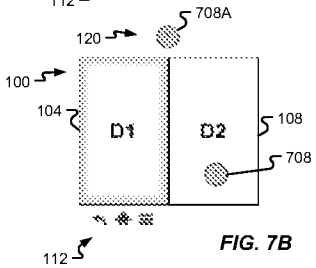


FIG. 7B

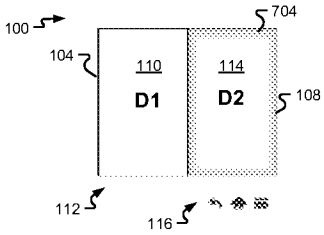


FIG. 7C

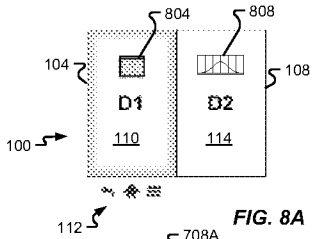


FIG. 8A

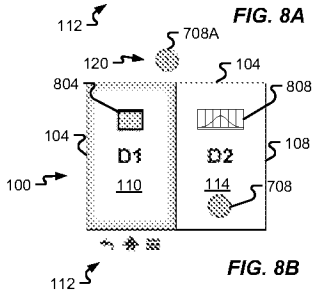


FIG. 8B

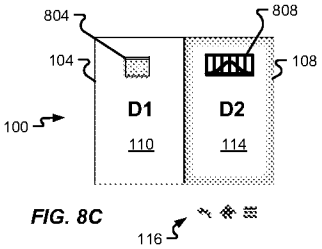


FIG. 8C

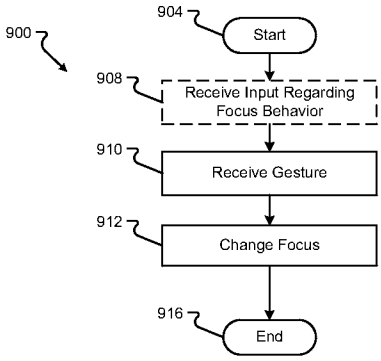


FIG. 9

[0100] FIGS. 7A-7C illustrate an embodiment showing device 100 and how device 100 responds when an incomplete gesture is made. FIG. 7A illustrates device 100 (with screens 104 and 108) when a focus is on screen 104 and display area 110. As noted above, "focus" means active and selected for receiving input. Accordingly, in FIG. 7A screen 104 and display area 110 are active and selected for input. Highlight 704 provides an indication to a user that screen 104 and display area 110 have the focus. If a user presses configurable area 112, device 100 will consider the inputs as inputs for screen 104 and/or for display 110.

[0101] As shown in FIG. 7B, a user can make a gesture 708, such as a drag or flick gesture on gesture capture area 120. Device 100 receives the gesture. A determination is made by device 100 as to whether the gesture 708 was complete. If the gesture was not complete, then as shown in FIG. 7C the focus is maintained on screen 104 and display area 110.

[0102] Highlighting 704 as noted above, indicates that screen 104 and display 110 are active. Highlighting 704 can be effected in a number of different ways. In some embodiments, the highlighting 704 is accomplished using light. In these embodiments, a backlight in device 100 may be configured to provide brighter light in some areas, such as around a border as shown in FIGS. 7A-7C. In other embodiments, the brighter light may be located on just one side/area or may be the entire display 110, which is made brighter than at least a portion of screen 108 and/or display 114.

[0103] In other embodiments, highlighting 704 may be accomplished using color, or a displayed image such as an icon that indicates that the screen 104 and display 110 have the focus. In other embodiments, the highlighting may be accomplished by some other difference, such as changing text to a different font type, size, or other characteristic. In some embodiments, the highlighting can be accomplished, or enhanced, by changing a different screen, display, and/or displayed image. For example, in embodiments, the screen, display and/or displayed image that does not have the focus, will be deemphasized by darkening, a color change, or other features that distinguishes it from the highlighted screen, display, and/or image. As can be appreciated, a combination of one or more of the above highlighting features may be used in some embodiments.

[0104] In one embodiment, in addition to highlighting 704, configurable area 112 of screen 104 is highlighted to indicate that screen 104 and display 110 have the focus. Device 100 can be configured so that configurable areas are only highlighted when the corresponding screen has the

focus. That is, configurable area 112 is highlighted when screen 104 is in focus, while configurable area 116 is highlighted when focus is on screen 108. As indicated above, the highlighting may be effected by lighting, color, or an image displayed near the configurable area. In the embodiment, shown in FIGS. 7A-7C, when screen 104 is in focus, the configurable area 112 is back lit to show buttons that can be used to receive input from a user.

[0105] In some embodiments, when configurable area 112 is highlighted, input can be received by pressing portions of area 112. At the same time, device 100 may be configured so that no input is received from configurable area 116. Thus, even if a user presses on area 116, no input will be received by device 100. Similarly, when configurable area 116 is highlighted device 100 will allow input to be received from area 116, and no input will be received from area 112. In other embodiments, only one configurable area will be highlighted at a time reflecting the particular screen, display, or image that is in focus. That is, when configurable area 112 is highlighted configurable area 116 will not be highlighted, and when configurable area 116 is highlighted configurable area 112 will not be highlighted.

[0106] It is noted that although gesture 708 is being made in gesture capture area 120, which is not on display 110, in other embodiments, the gesture 708 is made on one or more portions of displays 110 and 114. The gesture 708 may involve touching some portion of displays 110 and 114 (which are touch sensitive) or can involve only movement above the surface of displays 110 and 114.

[0107] In some embodiments, even though the gesture 708 is incomplete, device 100 may be configured to make changes to displays 104 and 108 to show that gesture 704 has been received. For example, in some embodiments image 712 (all of the image(s) displayed on display 110) appears to move across both displays 110 and 114 when gesture 708 is received. This feature allows a user to appreciate that the gesture 708 has been received, however it is probably incomplete as the focus was not changed to screen 108 and display 114 and was maintained on screen 104 and display 108 (FIG. 7C). The user can then make the gesture 708 again. This feature provides some visual feedback to a user regarding the incomplete gesture.

[0108] FIGS. 8A-8C illustrate another embodiment that provides feedback to a user about an incomplete gesture. FIG. 8A illustrates an image 804 displayed on display 110. The image 804 may be any type of displayed image such as a photo, a document, an icon that represent an application, or a window of an application. Image 804 is highlighted in FIG. 8A indicating that

it is in focus. When gesture 708 is received by gesture capture area 120, image 804 appears to move from display 110 to display 114. Because a determination is made that the gesture was not complete, the image 804 is returned back to its original position. The movement of image 804 indicates to a user that the gesture 708 was received but because image 804 was not displayed on display 114 it must have been incomplete. The user can then make gesture 708 again to change focus.

[0109] In some embodiments, image 712 or image 804 may be displayed as moving depending upon how device 100 is configured. For example, the device 100 may be configured to move image 712 (entire image(s) displayed on display 110) when the gesture 708 is a drag gesture and image 804 (active window, icon, photo, document) if gesture 708 is a flick gesture or some other gesture. In other embodiments, the location of the gesture 708, whether in gesture capture region 120 or some other portion of displays 110 and 114, may determine whether image 710 or image 804 is shown as moving from display 110 to display 114.

[0110] FIGS. 9A-9C illustrates an embodiment of device 100 and its output in response to receiving a completed gesture. In FIG. 9A the focus is on screen 104, display 110, and image 804. Image 804 is in embodiments a window of an active application which is selected to receive input from a user. In the embodiment shown in FIGS. 9A-9C, a user makes a gesture 708 in gesture capture region 120. The gesture is a complete gesture. In response to the complete gesture, device 100 changes focus away from screen 104, display 110, and image 804 and to screen 108 and display 114.

[0111] In some embodiments, as shown in FIG. 9C, in addition to changing the focus to screen 108 and display 114, the focus is also automatically on an image that is displayed on display 114, i.e., image 904. In these embodiments, device 100 may be configured by a user so that when focus is changed to a screen or display, the focus will also be on an image, e.g., a window of an open application, if one is open on the display to which the focus has changed. The user may, for example, list applications in order of priority such that if there is an open window the focus will change to an application at the top of the list. If no window is open for an application at the top the list, then the list can be traversed in order of priority until an application with an open window is found. If none of the applications on the list have open windows then the focus will not change to any images on the display. In some embodiments, the focus may be changed to a window of the same application that was previously in focus.

[0112] In other embodiments, the focus may always change to a default image (e.g., open window) selected by a user. In some embodiments, if the default application does not have an open window, device 100 may be configured to launch the application and open a window that will then be in focus.

[0113] In some embodiments, device 100 is configured so that depending upon the particular gesture 708, an image on the newly focused display may be in focus or not. For example, the device 100 may be configured to have in image in focus when the gesture 708 is a drag gesture. If the gesture 708 is a flick gesture or some other gesture, no image will be in focus, and instead only display 114 will be focus. In other embodiments, the location of the gesture 708, whether in gesture capture region 120 or some other portion of displays 110 and 114, may determine whether an image on the newly focused display may be in focus or not.

[0114] It is noted that although gesture 708 is being made in gesture capture area 120, which is not on display 110 or display 114, in other embodiments, the gesture 708 is made on one or more portions of displays 110 and 114. The gesture 708 may involve touching some portion of displays 110 and 114 (which are touch sensitive) or can involve only movement above the surface of displays 110 and 114.

[0115] As shown in FIG. 9C, when focus is changed to screen 108 and display 114, configurable area 116 is highlighted to indicate that the focus is changed. Device 100 can be configured so that configurable areas are only highlighted when the corresponding screen has the focus. That is, configurable area 112 is highlighted when screen 104 and/or display 110 is in focus, while configurable area 116 is highlighted when focus is on screen 108 and/or display 114. As indicated above, the highlighting may be effected by lighting, color, or an image displayed near the configurable area. In the embodiment, shown in FIGS. 9A-9C, when screen 104 and/or display 110 is in focus, the configurable area 112 is back lit to show buttons that can be used to receive input from a user. Similarly, when focus changes to screen 108 and/or display 114, the configurable area 116 is back lit to show buttons that can be used to receive input from a user.

[0116] Referring now to FIG. 10, a flow diagram 1000 in accordance with at least some embodiments of the present disclosure is shown and will be described. Flow 1000 is in embodiments performed by a device such as device 100. More specifically, one or more hardware or software components may be involved in performing flow 1000. In one

embodiment, modules in middleware 520 (FIG. 5A) such as multi-display management class 524 (FIG. 5B) perform one or more of the steps of flow 1000. In other embodiments, in addition to, or in lieu of, middleware 520 performing steps of flow 1000, operating system kernel(s) 516a, processor 504, and/or drivers 512a-512b may also perform steps of flow 1000.

[0117] Flow 1000 is initiated at 1004. Flow 1000 passes from 1004 to optional step 1008 where input regarding focus behavior is received. In this step, a user may provide configuration information that indicates how a device should handle focus when receiving gestures. In embodiments, the device is device 100 (FIGS. 1-3) which includes multiple screens. The user may input any information that affects how a device 100 may handle input gestures, including but not limited to, changes (if any) that occur to displays 104 and 108 when gestures are incomplete to show that a gesture has been received, what is highlighted when focus is changed, priority list of applications when focus is changed, whether focus changes are limited to a display, screen, and/or displayed image, whether focus behavior changes based on the type of gesture received, and other information. In one embodiment, the input indicates that in response to an incomplete gesture, the focus should be maintained on the screen, display, and/or image with the focus.

[0118] From step 1008, flow 1000 passes to step 1012 where a gesture is received. The gesture can be received in some embodiments by a gesture capture area such as areas 120 and 124 and/or pressure sensitive displays 110 and 114. In some embodiments, as shown in FIGS. 1 and 2, the gesture capture area(s) is distinct and located away from the pressure sensitive displays. In other embodiments the gesture capture area may overlap partially or completely with the pressure sensitive displays.

[0119] The gesture may be any type of gesture. In embodiments, the gesture is one or more of the gestures illustrated in FIG. 4A-4H. The device 100 is configured so that the gesture received at step1012 indicates a request to change focus from the current focus.

[0120] In some embodiments, flow 1000 may include an optional step 1016 that changes one or more of the displays to indicate the receipt of the gesture. The change may be for example, moving an image displayed on a first screen toward a second screen. The particular changes made to the displays may be based, in part, on the information received at optional step 1008.

[0121] After step 1016, a determination is made at 1020 as to whether the gesture is complete. A determination can be made that the gesture is complete if the input is recognized as a complete

gesture. In other words, data at step 1012 corresponds to a gesture stored within the memory of device 100. If, however, the data received at step 1012 does not correspond to any gesture stored within the memory of device 100, then a determination can be made that the gesture is not complete.

[0122] In those embodiments in which the gesture is complete, flow 1000 passes from determination 1020 to step 1024 where the focus is changed. The focus is changed in accordance with predetermined configurations, some of which may be based on the input received at optional step 1008. For example, if the device 100 is configured so that the gesture received at step 1012 indicates that the focus should be changed to a second display, at step 1024 the focus will be changed to the second display. If the configuration information indicates that the focus should also be changed to an open window displayed on the second display, then at step 1024 the open window displayed on the second display will also be in focus. Afrer step 1024, flow 1000 ends at 1028.

[0123] It should be noted that in some embodiments, step 1024 may involve a number of sub steps. For example, in some embodiments as part of step 1024, a configurable area may be highlighted, a display may be highlighted, an icon may be generated indicating the focus change, and/or portions of a first screen may be changed to emphasize the highlighting on a second screen. In some embodiments, the substeps may involve not allowing input from a configurable area associated with a first screen and allowing input from a configurable area associated with a second screen.

[0124] If at determination 1020, it is determined that the gesture is not complete, then flow passes to step 1032. At step 1032, the focus is maintained on the screen, display, and/or image where it is currently. Flow 1000 ends at 1028.

[0125] The exemplary systems and methods of this disclosure have been described in relation to FIGS 1-10. However, to avoid unnecessarily obscuring the present disclosure, the preceding description omits a number of known structures and devices. This omission is not to be construed as a limitation of the scopes of the claims. Specific details are set forth to provide an understanding of the present disclosure. It should however be appreciated that the present disclosure may be practiced in a variety of ways beyond the specific detail set forth herein.

[0126] Furthermore, while the exemplary aspects, embodiments, and/or configurations illustrated herein show the various components of the system collocated, certain components of

the system can be located remotely, at distant portions of a distributed network, such as a LAN and/or the Internet, or within a dedicated system. Thus, it should be appreciated, that the components of the system can be combined in to one or more devices, such as a _____, or collocated on a particular node of a distributed network, such as an analog and/or digital telecommunications network, a packet-switch network, or a circuit-switched network. It will be appreciated from the preceding description, and for reasons of computational efficiency, that the components of the system can be arranged at any location within a distributed network of components without affecting the operation of the system. For example, the various components can be located in a switch such as a PBX and media server, gateway, in one or more communications devices, at one or more users' premises, or some combination thereof. Similarly, one or more functional portions of the system could be distributed between a telecommunications device(s) and an associated computing device.

[0127] Furthermore, it should be appreciated that the various links connecting the elements can be wired or wireless links, or any combination thereof, or any other known or later developed element(s) that is capable of supplying and/or communicating data to and from the connected elements. These wired or wireless links can also be secure links and may be capable of communicating encrypted information. Transmission media used as links, for example, can be any suitable carrier for electrical signals, including coaxial cables, copper wire and fiber optics, and may take the form of acoustic or light waves, such as those generated during radio-wave and infra-red data communications.

[0128] Also, while the flowcharts have been discussed and illustrated in relation to a particular sequence of events, it should be appreciated that changes, additions, and omissions to this sequence can occur without materially affecting the operation of the disclosed embodiments, configuration, and aspects.

[0129] A number of variations and modifications of the disclosure can be used. It would be possible to provide for some features of the disclosure without providing others.

[0130] For example in one alternative embodiment, all of the focus behavior information may be preprogrammed into the device 100. In these embodiments, there would be no input received by a user regarding how a device 100 handles gestures indicating a change in focus.

[0131] In other alternative embodiments, device 100 may be in any configuration described above with respect to FIGS. 6A-6J. In these embodiments, gestures would be received as

described above indicating a request for changing focus. The focus would then change from a first screen, display, or image to a second screen, display, or image.

[0132] In yet another embodiment, the systems and methods of this disclosure can be implemented in conjunction with a special purpose computer, a programmed microprocessor or microcontroller and peripheral integrated circuit element(s), an ASIC or other integrated circuit, a digital signal processor, a hard-wired electronic or logic circuit such as discrete element circuit, a programmable logic device or gate array such as PLD, PLA, FPGA, PAL, special purpose computer, any comparable means, or the like. In general, any device(s) or means capable of implementing the methodology illustrated herein can be used to implement the various aspects of this disclosure. Exemplary hardware that can be used for the disclosed embodiments, configurations and aspects includes computers, handheld devices, telephones (e.g., cellular, Internet enabled, digital, analog, hybrids, and others), and other hardware known in the art. Some of these devices include processors (e.g., a single or multiple microprocessors), memory, nonvolatile storage, input devices, and output devices. Furthermore, alternative software implementations including, but not limited to, distributed processing or component/object distributed processing, parallel processing, or virtual machine processing can also be constructed to implement the methods described herein.

[0133] In yet another embodiment, the disclosed methods may be readily implemented in conjunction with software using object or object-oriented software development environments that provide portable source code that can be used on a variety of computer or workstation platforms. Alternatively, the disclosed system may be implemented partially or fully in hardware using standard logic circuits or VLSI design. Whether software or hardware is used to implement the systems in accordance with this disclosure is dependent on the speed and/or efficiency requirements of the system, the particular function, and the particular software or hardware systems or microprocessor or microcomputer systems being utilized.

[0134] In yet another embodiment, the disclosed methods may be partially implemented in software that can be stored on a storage medium, executed on programmed general-purpose computer with the cooperation of a controller and memory, a special purpose computer, a microprocessor, or the like. In these instances, the systems and methods of this disclosure can be implemented as program embedded on personal computer such as an applet, JAVA® or CGI script, as a resource residing on a server or computer workstation, as a routine embedded in a

dedicated measurement system, system component, or the like. The system can also be implemented by physically incorporating the system and/or method into a software and/or hardware system.

[0135] Although the present disclosure describes components and functions implemented in the aspects, embodiments, and/or configurations with reference to particular standards and protocols, the aspects, embodiments, and/or configurations are not limited to such standards and protocols. Other similar standards and protocols not mentioned herein are in existence and are considered to be included in the present disclosure. Moreover, the standards and protocols mentioned herein and other similar standards and protocols not mentioned herein are periodically superseded by faster or more effective equivalents having essentially the same functions. Such replacement standards and protocols having the same functions are considered equivalents included in the present disclosure.

[0136] The present disclosure, in various aspects, embodiments, and/or configurations, includes components, methods, processes, systems and/or apparatus substantially as depicted and described herein, including various aspects, embodiments, configurations embodiments, subcombinations, and/or subsets thereof. Those of skill in the art will understand how to make and use the disclosed aspects, embodiments, and/or configurations afrer understanding the present disclosure. The present disclosure, in various aspects, embodiments, and/or configurations, includes providing devices and processes in the absence of items not depicted and/or described herein or in various aspects, embodiments, and/or configurations hereof, including in the absence of such items as may have been used in previous devices or processes, e.g., for improving performance, achieving ease and\or reducing cost of implementation.

[0137] The foregoing discussion has been presented for purposes of illustration and description. The foregoing is not intended to limit the disclosure to the form or forms disclosed herein. In the foregoing Detailed Description for example, various features of the disclosure are grouped together in one or more aspects, embodiments, and/or configurations for the purpose of streamlining the disclosure. The features of the aspects, embodiments, and/or configurations of the disclosure may be combined in alternate aspects, embodiments, and/or configurations other than those discussed above. This method of disclosure is not to be interpreted as reflecting an intention that the claims require more features than are expressly recited in each claim. Rather, as the following claims reflect, inventive aspects lie in less than all features of a single foregoing

disclosed aspect, embodiment, and/or configuration. Thus, the following claims are hereby incorporated into this Detailed Description, with each claim standing on its own as a separate preferred embodiment of the disclosure.

[0138] Moreover, though the description has included description of one or more aspects, embodiments, and/or configurations and certain variations and modifications, other variations, combinations, and modifications are within the scope of the disclosure, e.g., as may be within the skill and knowledge of those in the art, afrer understanding the present disclosure. It is intended to obtain rights which include alternative aspects, embodiments, and/or configurations to the extent permitted, including alternate, interchangeable and/or equivalent structures, functions, ranges or steps to those claimed, whether or not such alternate, interchangeable and/or equivalent structures, functions, ranges or steps are disclosed herein, and without intending to publicly dedicate any patentable subject matter.

What is claimed is:

1. A method, comprising:

receiving, by at least one of a gesture capture region and a touch sensitive display, a gesture indicating a request to change focus;

determining by at least one processor whether the gesture is completed; and

in response to the at least one processor determining that the gesture is not completed, maintaining a focus on a first screen that has the focus.

2. The method of claim 1, further comprising:

in response to the at least one processor determining that the gesture is completed, changing the focus from the first screen to a second screen.

3. The method of claim 1, further comprising:

in response to the determining that the gesture is completed, further changing the focus from a first image displayed on the first screen to a second image displayed on a second screen.

4. The method of claim 3, further comprising:

wherein the first displayed image is a displayed window and the second displayed image is a second window.

5. The method of claim 1, further comprising:

receiving input indicating that in response to an incomplete gesture, the focus should be maintained on the screen with the focus.

6. The method of claim 1, further comprising:

highlighting a configurable area on the first screen when the focus is on the screen; and

highlighting a second configurable area on the second screen when the focus is on the second screen.

7. The method of claim 1, further comprising: in response to receiving the gesture, displaying at least one image displayed on the first screen as moved from an original position.

8. The method of claim 7, in response to receiving the gesture, displaying the at least one image as moved in a direction of the gesture.

9. The method of claim 8, further comprising:

in response to the determining that the gesture is incomplete, displaying the at least one image in the original position.

10. A non-transitory computer readable medium storing computer executable instructions that when executed by at least one processor perform a method comprising:

receiving, by a gesture capture region, a drag gesture indicating a request to change focus;

determining by at least one processor whether the drag gesture is completed;

in response to the at least one processor determining that the drag gesture is not completed, maintaining a focus on an image displayed on a first screen; and

in response to the at least one processor determining that the gesture is completed, changing the focus from the first image to a second image displayed on a second screen.

11. The non-transitory computer readable medium of claim 10, wherein the method further comprises:

in response to the at least one processor determining that the drag gesture is not completed, maintaining the lighting of a portion of a first configurable area of the first screen and allowing input from the first configurable area.

12. The non-transitory computer readable medium of claim 11, wherein the method further comprises:

allowing input from the first configurable area.

13. The non-transitory computer readable medium of claim 10, wherein the method further comprises:

in response to the at least one processor determining that the gesture is completed, darkening the portion of the first configurable area of the first screen and lighting up a portion of a second configurable area of the second screen.

14. The non-transitory computer readable medium of claim 13, wherein the method further comprises:

allowing input from the second configurable area; and

not allowing input from the first configurable area.

15. The non-transitory computer readable medium of claim 10, wherein the method further comprises:

receiving input indicating that in response to an incomplete gesture, the focus should be maintained on a displayed image with the focus.

16. The non-transitory computer readable medium of claim 10, wherein the first displayed image is a first window and the second displayed image is a second window.

17. A dual screen communication device, comprising:

a gesture capture region to receive a gesture; and

a first touch sensitive display;

a second touch sensitive display;

a computer readable medium that stores computer executable instructions that when executed by at least one processor perform a method comprising:

receiving, by one or more of the gesture capture region, the first touch sensitive display, and the second touch display, a gesture indicating a request to change focus;

determining by at least one processor whether the gesture is completed; and

in response to the at least one processor determining that the gesture is not completed, maintaining a focus on the first touch sensitive display.

18. The device of claim 17, wherein the method further comprises:

in response to the at least one processor determining that the gesture is completed, changing the focus from the first touch sensitive display to a second touch sensitive display.

19. The device of claim 17, wherein the method further comprises:

in response to the at least one processor determining that the drag gesture is not completed, allowing input from the first touch sensitive display and not allowing input from a portion of the second touch sensitive display.

20. The device of claim 17, wherein the method further comprises:

in response to the at least one processor determining that the drag gesture is completed, allowing input from the second touch sensitive display and not allowing input from a portion of the first touch sensitive display.

ABSTRACT

Embodiments are described for handling focus when a gesture is input in a multi-screen device. In embodiments, if the gesture is incomplete a focus is maintained on a screen, display area, or displayed image that has the focus. If the gesture is determined to be complete, then the focus is changed to a different screen, display area, or displayed image.

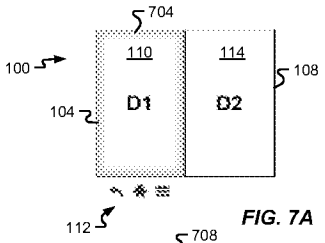


FIG. 7A

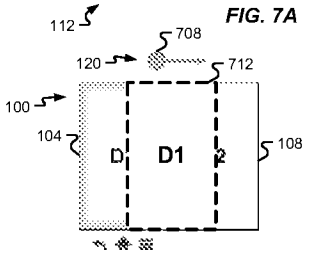


FIG. 7B

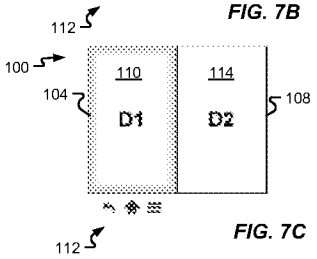


FIG. 7C

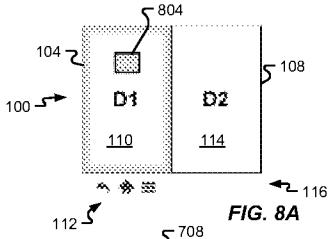


FIG. 8A

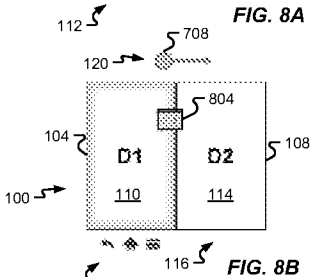


FIG. 8B

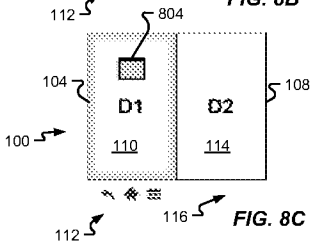


FIG. 8C

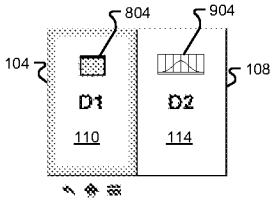


FIG. 9A

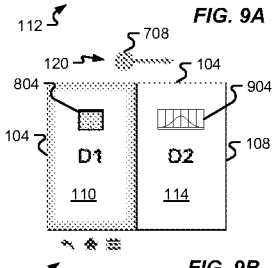


FIG. 9B

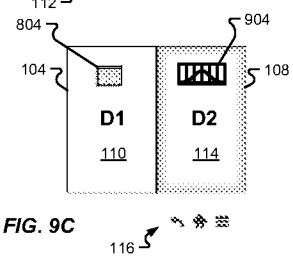


FIG. 9C

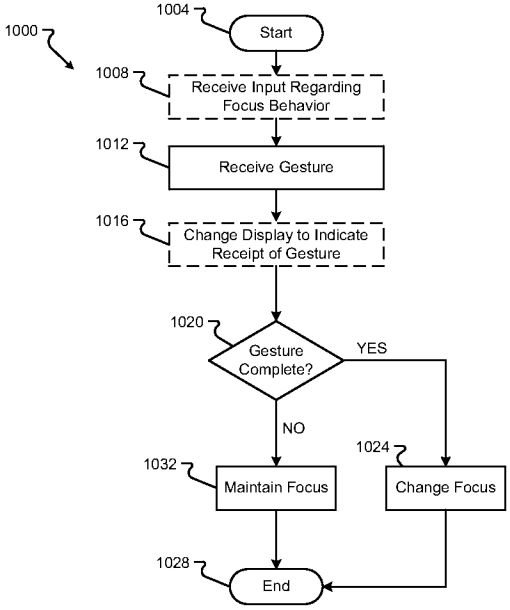


FIG. 10

HARDWARE BUTTONS ACTIVATED BASED ON FOCUS

[0001] Figs. 7A and 7B depict a device 100 with a first screen 104 and a second screen 108 in a portrait orientation. More particularly, in the operating mode illustrated in Fig. 7A, the first screen 104 includes a touch sensitive display 110 that has a current focus. Moreover, because the first touch sensitive display 110 has the current focus, a set of control buttons 704 is provided in a configurable area 112 of the first screen 104. More particularly, as depicted in the figure, the control buttons 704 can include, as examples and without limitation, a first area 112a comprising a "back" input button, a second area 112b comprising a "menu" input button, and a third area 112c comprising a "home" input button. In addition, it can be seen that the second screen 108 does not include control buttons 704 in the configurable area 116 associated with the second screen 108, when, as in the example of Fig. 7A, the current focus is on the first touch sensitive display 110 or information contained therein.

[0100] In Fig. 7B, the focus is on the second screen 108. Accordingly, the configurable area 116 of the second screen 108 includes a set of control buttons 704. Moreover, with the focus on the second screen 108, and in particular on the second touch sensitive display 114, the first screen 104 does not include control buttons 704 within the configurable area 112 of the first screen 104.

[0101] In accordance with further embodiments of the present invention, configurable buttons 704 can be selectively provided in association with a screen 104 or 108 having a current focus when the screens 104, 108 are in a landscape orientation. As illustrated in Fig. 8A, in a landscape orientation, a set of control buttons 704 is associated with the first screen 104, when the focus is on the first screen. In particular, the focus can be on all or a portion of the application or information contained within the first touch sensitive display 110 of the first screen 104. In addition, with the focus on the first screen 104, there are no control buttons associated with the configurable area 112 of the second screen 108.

[0102] In Fig. 8B, a device 100 is shown with the first 104 and second 108 screens in a landscape orientation, with the current focus on the second screen 108. In this configuration, control buttons 704 are illustrated as part of the second screen 108. Moreover, there are no control buttons provided as part of the first screen 104. Although the examples in Figs. 8A and 8B illustrate configurable buttons 704 on the left side of a screen 104 or 108 having a current focus, the control buttons 704 can alternatively be on the right side of the screens 104, 108 when

those screens 104, 108 are in a landscape orientation. In particular, in embodiments in which the control buttons 704 are presented within a configurable area 112, 116, whether the control buttons 704 appear on the left or right hand side of a screen 104, 108 when the screen 104, 108 is in a landscape mode depends on whether the configurable area 112, 116 is on the left or right hand side of the screen 104, 108.

[0103] With reference now to Fig. 9, aspects of a method for providing hardware buttons activated based on focus in accordance with embodiments of the present invention are depicted. More particularly, methods for providing hardware buttons in the form of one or more control buttons as part of a screen 104, 108 of a device 100 are depicted. Initially, the system is started, and output is provided using the first 104 and second 108 screens of the device 100 (step 904). Accordingly, the device 100 is in an output configuration in which both displays 104, 108 are operative. Moreover, the screens 104, 108 can be oriented in a dual portrait configuration (*see, e.g.*, Figs. 6C, 7A and 7B), or a dual landscape output configuration (*see, e.g.*, Figs. 6F, 8A and 8B).

[0104] At step 908, a determination is made as to which screen 104, 108 currently has focus. In accordance with embodiments of the present invention, the screen 104, 108 with focus can be determined from the screen 104, 108 containing an active application, application window, or requested application. In accordance with still other embodiments, the screen that currently has focus can be determined from input provided by the user. For instance, by tapping a screen 104, 108 within a touch sensitive region 110, 112, 114, 116, 120, 124 of that screen 104, 108, the user can indicate the screen 104, 108 that has focus. In accordance with embodiments of the present invention, a screen with focus is the screen that is currently enabled for receiving input. In accordance with other embodiments, the screen with the current focus is operated as the primary screen, while the other screen is operated as a secondary screen.

[0105] One or more control buttons, for example a set of control buttons 704, is displayed on the screen 104, 108 that has the current focus (step 912). In accordance with embodiments of the present invention, the control buttons 704 can be presented within a configurable area 112, 116 of the screen 104, 108 that has the current focus. Alternatively, one or more control buttons can be presented by other touch sensitive areas of the screen 104, 108 with the current focus, for example within a touch sensitive display 110, 114 area. In accordance with embodiments of the present invention, control buttons 704 are only displayed on the screen 104 or 108 that has the

current focus. The screen 104 or 108 that does not have the current focus does not include control buttons 704.

[0106] At step 916, a determination is made as to whether the focus has changed. In particular, a determination can be made as to whether the current focus has shifted from a first one of the screens 104, 108 to a second one of the screens 104, 108. If the focus has shifted, the process can return to step 908 to determine which screen has the current focus. Alternatively, for example where the device 100 has only a first 104 and second 108 screen, the process may proceed directly to step 912, as any change in focus that is detected indicates that the other of the two screens 104, 108 has the current focus after the change. As with an initial selection of a screen 104, 108 with a current focus, a change in focus can be indicated as a result of the operation of application programming, or as a result of input received from the user.

[0107] At step 920, a determination can be made as to whether the device 100 has been powered off. If the device 100 has not been powered off, the process can return to step 904, and output can continue to be provided by the screens 104, 108. If the device 100 has been powered off, the process may end. Although a particular progression of steps has been illustrated, it should be appreciated that other sequences of operation are possible. For example, at any point within the depicted process, providing output can be discontinued of the device 100 is powered off. In addition, the described process is generally not performed when the device 100 is in a configuration in which only one screen 104, 108 is operative. In particular, where only one touch screen 104, 108 is operative, one or more control buttons 704 can be presented by that operative screen 104, 108.

[0108] In accordance with embodiments of the present invention, the method disclosed herein can be performed by the execution of application programming stored in memory 208, 508 by a processor 204, 504. For instance, a windows management module or class 532 can include functionality to identify the screen 104, 108 that has the current focus, and to provide control buttons 704 in association with the screen 104, 108 having focus. In accordance with still other embodiments, a component or components of the framework 520, such as the window management class 532, can operate to receive input from a user indicating a selection of a screen 104, 108 to which the current focus has moved, and to present one or more control buttons 704 in association with the screen 104, 108 that has focus.

[0109] As described herein, the current focus can be determined in consideration of the screen 104, 108 presenting an active application, a launched application, or a primary screen. Alternatively or in addition, the current focus can be determined in response to input received from the user. As an example, and without limitation, such user input can be a touch input entered within a touch sensitive area of a screen 104, 108, such as a touch sensitive display area 110, 114, a configurable area 112, 116, and/or a gesture capture region 120, 124.

[0110] The exemplary systems and methods of this disclosure have been described in relation to configurable hardware buttons. However, to avoid unnecessarily obscuring the present disclosure, the preceding description omits a number of known structures and devices. This omission is not to be construed as a limitation of the scopes of the claims. Specific details are set forth to provide an understanding of the present disclosure. It should however be appreciated that the present disclosure may be practiced in a variety of ways beyond the specific detail set forth herein.

[0111] Furthermore, while the exemplary aspects, embodiments, and/or configurations illustrated herein show the various components of the system collocated, certain components of the system can be located remotely, at distant portions of a distributed network, such as a LAN and/or the Internet, or within a dedicated system. Thus, it should be appreciated, that the components of the system can be combined in to one or more devices, such as a computer, laptop, netbook, tablet computer, smart phone, mobile device, etc., or collocated on a particular node of a distributed network, such as an analog and/or digital telecommunications network, a packet-switch network, or a circuit-switched network. It will be appreciated from the preceding description, and for reasons of computational efficiency, that the components of the system can be arranged at any location within a distributed network of components without affecting the operation of the system. For example, the various components can be located in a switch such as a PBX and media server, gateway, in one or more communications devices, at one or more users' premises, or some combination thereof. Similarly, one or more functional portions of the system could be distributed between a telecommunications device(s) and an associated computing device.

[0112] Furthermore, it should be appreciated that the various links connecting the elements can be wired or wireless links, or any combination thereof, or any other known or later developed element(s) that is capable of supplying and/or communicating data to and from the connected

elements. These wired or wireless links can also be secure links and may be capable of communicating encrypted information. Transmission media used as links, for example, can be any suitable carrier for electrical signals, including coaxial cables, copper wire and fiber optics, and may take the form of acoustic or light waves, such as those generated during radio-wave and infra-red data communications.

[0113] Also, while the flowcharts have been discussed and illustrated in relation to a particular sequence of events, it should be appreciated that changes, additions, and omissions to this sequence can occur without materially affecting the operation of the disclosed embodiments, configuration, and aspects.

[0114] A number of variations and modifications of the disclosure can be used. It would be possible to provide for some features of the disclosure without providing others.

[0115] For example in one alternative embodiment, in a desktop reveal operation, where an active application is moved off screen to reveal a desktop or portion of a desktop, the screen 104, 108 that contained the active application can retain the focus.

[0116] In another alternative embodiment, the control keys 704 can be hardware keys that are selectively activated, or keys presented by operation of a configurable area 112, 116 comprising a touch screen display capable of displaying keys or icons comprising the control buttons 704 as part of a touch screen display.

[0117] In yet another embodiment, the systems and methods of this disclosure can be implemented in conjunction with a special purpose computer, a programmed microprocessor or microcontroller and peripheral integrated circuit element(s), an ASIC or other integrated circuit, a digital signal processor, a hard-wired electronic or logic circuit such as discrete element circuit, a programmable logic device or gate array such as PLD, PLA, FPGA, PAL, special purpose computer, any comparable means, or the like. In general, any device(s) or means capable of implementing the methodology illustrated herein can be used to implement the various aspects of this disclosure. Exemplary hardware that can be used for the disclosed embodiments, configurations and aspects includes computers, handheld devices, telephones (e.g., cellular, Internet enabled, digital, analog, hybrids, and others), and other hardware known in the art. Some of these devices include processors (e.g., a single or multiple microprocessors), memory, nonvolatile storage, input devices, and output devices. Furthermore, alternative software implementations including, but not limited to, distributed processing or component/object

distributed processing, parallel processing, or virtual machine processing can also be constructed to implement the methods described herein.

[0118] In yet another embodiment, the disclosed methods may be readily implemented in conjunction with software using object or object-oriented software development environments that provide portable source code that can be used on a variety of computer or workstation platforms. Alternatively, the disclosed system may be implemented partially or fully in hardware using standard logic circuits or VLSI design. Whether software or hardware is used to implement the systems in accordance with this disclosure is dependent on the speed and/or efficiency requirements of the system, the particular function, and the particular software or hardware systems or microprocessor or microcomputer systems being utilized.

[0119] In yet another embodiment, the disclosed methods may be partially implemented in software that can be stored on a storage medium, executed on programmed general-purpose computer with the cooperation of a controller and memory, a special purpose computer, a microprocessor, or the like. In these instances, the systems and methods of this disclosure can be implemented as program embedded on personal computer such as an applet, JAVA® or CGI script, as a resource residing on a server or computer workstation, as a routine embedded in a dedicated measurement system, system component, or the like. The system can also be implemented by physically incorporating the system and/or method into a software and/or hardware system.

[0120] Although the present disclosure describes components and functions implemented in the aspects, embodiments, and/or configurations with reference to particular standards and protocols, the aspects, embodiments, and/or configurations are not limited to such standards and protocols. Other similar standards and protocols not mentioned herein are in existence and are considered to be included in the present disclosure. Moreover, the standards and protocols mentioned herein and other similar standards and protocols not mentioned herein are periodically superseded by faster or more effective equivalents having essentially the same functions. Such replacement standards and protocols having the same functions are considered equivalents included in the present disclosure.

[0121] The present disclosure, in various aspects, embodiments, and/or configurations, includes components, methods, processes, systems and/or apparatus substantially as depicted and described herein, including various aspects, embodiments, configurations embodiments,

subcombinations, and/or subsets thereof. Those of skill in the art will understand how to make and use the disclosed aspects, embodiments, and/or configurations after understanding the present disclosure. The present disclosure, in various aspects, embodiments, and/or configurations, includes providing devices and processes in the absence of items not depicted and/or described herein or in various aspects, embodiments, and/or configurations hereof, including in the absence of such items as may have been used in previous devices or processes, e.g., for improving performance, achieving ease and/or reducing cost of implementation.

[0122] The foregoing discussion has been presented for purposes of illustration and description. The foregoing is not intended to limit the disclosure to the form or forms disclosed herein. In the foregoing Detailed Description for example, various features of the disclosure are grouped together in one or more aspects, embodiments, and/or configurations for the purpose of streamlining the disclosure. The features of the aspects, embodiments, and/or configurations of the disclosure may be combined in alternate aspects, embodiments, and/or configurations other than those discussed above. This method of disclosure is not to be interpreted as reflecting an intention that the claims require more features than are expressly recited in each claim. Rather, as the following claims reflect, inventive aspects lie in less than all features of a single foregoing disclosed aspect, embodiment, and/or configuration. Thus, the following claims are hereby incorporated into this Detailed Description, with each claim standing on its own as a separate preferred embodiment of the disclosure.

[0123] Moreover, though the description has included description of one or more aspects, embodiments, and/or configurations and certain variations and modifications, other variations, combinations, and modifications are within the scope of the disclosure, e.g., as may be within the skill and knowledge of those in the art, after understanding the present disclosure. It is intended to obtain rights which include alternative aspects, embodiments, and/or configurations to the extent permitted, including alternate, interchangeable and/or equivalent structures, functions, ranges or steps to those claimed, whether or not such alternate, interchangeable and/or equivalent structures, functions, ranges or steps are disclosed herein, and without intending to publicly dedicate any patentable subject matter.

What is claimed is:

1. A method for presenting control buttons on a device, comprising:

providing a first output on a first screen of a device;

providing a second output on a second screen of the device;

determining which one of the first and second screens contains information that currently has focus;

displaying at least a first control button in association with the screen determined to contain information that currently has focus.

2. The method of claim 1, wherein determining which of the screens contains information that currently has focus includes determining which of the screens contains information that has been selected by a user.

3. The method of claim 1, wherein determining which of the screens contains information that currently has focus includes applying rules to determine which of the screens contains information that currently has focus.

4. The method of claim 1, wherein the first output on the first screen includes a display of information associated with a first application, wherein the first application is determined to have focus, wherein the first control button is presented by the first screen.

5. The method of claim 4, wherein the second output on the second screen includes a display of information associated with the first application, and wherein the first screen is a primary screen.

6. The method of claim 4, wherein the second output on the second screen includes a display of information including at least a first item of information, wherein user input is received selecting that at least a first item of information, wherein the focus is shifted from the first screen to the second screen, and wherein in response to the shift in focus the first control button is presented by the second screen.

7. The method of claim 6, wherein the display of information including the first item of information on the second screen is associated with a second application.

8. The method of claim 4, further comprising:

receiving a selection of a second application;

displaying information associated with the second application on the second screen;

shifting focus from the first screen to the second screen, wherein in response to the shift in focus the first control button is presented by the second screen.

9. The method of claim 1, wherein determining which one of the first and second screen contains information that currently has focus is determined in response to receiving input from a user in the form of a gesture.

10. The method of claim 9, wherein the gesture is received in a gesture capture region of one of the first screen and the second screen.

11. The method of claim 9, wherein the gesture spans portions of the first and second screens.

12. The method of claim 1, wherein after a first step of determining the first control button is displayed by the first screen and not by the second screen, wherein after a second step of determining the first control button is displayed by the second screen and not by the first screen.

13. A multiple screen device, comprising:

a first screen;

a second screen;

memory;

a processor;

application programming stored on the memory and executed by the processor, wherein the application programming is operable to identify one of the first and second screens as a screen displaying information having a current focus, and wherein the application programming is operable to display at least a first control button on the identified one of the first and second screens displaying information having a current focus.

14. The device of Claim 13, wherein the first and second screens each include:

a touch sensitive display area;

a touch sensitive configurable area, wherein the at least a first control button is displayed as part of the configurable area of the one of the first and second screens identified as displaying information having a current focus.

15. The device of claim 14, wherein the application programming is operable to identify one of the first and second screens as the screen displaying information having a current focus in response to touch input received at one of the touch sensitive display area or the touch sensitive configurable area of the identified one of the first and second screens.

16. The device of claim 14, wherein the first and second screens each include:

a gesture capture area, wherein the application programming is operable to identify one of the first and second screens as the screen displaying information having a current focus in response to touch input received at the touch sensitive gesture capture area of the identified one of the first and second screens.

17. A computer readable medium having stored theron computer-executable instructions, the computer executable instructions causing a processor to execute a method for presenting a control button, the computer executable instructions comprising:

instructions to display information on first and second screens;

instructions to identify one of the first and second screens having a current focus;

instructions to display the control button on the identified one of the first and second screens having a current focus.

18. The computer readable medium of claim 17, the computer readable instructions further comprising:

instructions to receive input from a touch screen area of one of the first and second screens, wherein the touch screen receiving the input is identified as the one of the first and second screens having a current focus.

19. The computer readable medium of claim 17, wherein the control button is displayed within a configurable area.

20. The computer readable medium of claim 17, wherein a first window related to a first application is displayed on the first screen, wherein the first window is an active window, wherein the first screen is identified as having a current focus, and wherein the control button is displayed on the first screen.

ABSTRACT

Methods and devices for providing one or more control buttons in connection with a multiple screen device are provided. More particularly, the screen of a multiple screen device having a current focus is identified, and one or more control buttons are provided as part of or in association with the identified screen. When a change in focus from a previously identified screen to a different screen is detected, the presentation of the one or more control buttons can also change. In particular, control buttons are presented as part of a screen having the current focus, while control buttons are not provided on or in association with a screen that does not have the current focus.

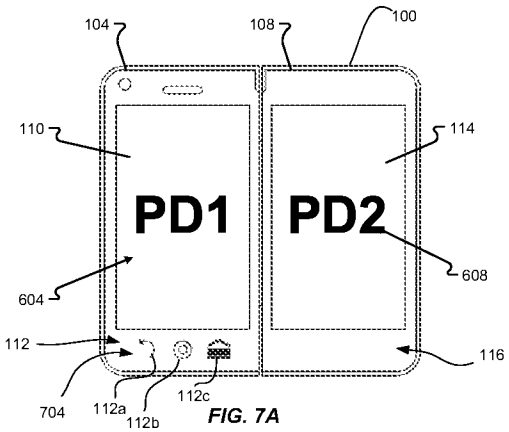


FIG. 7A

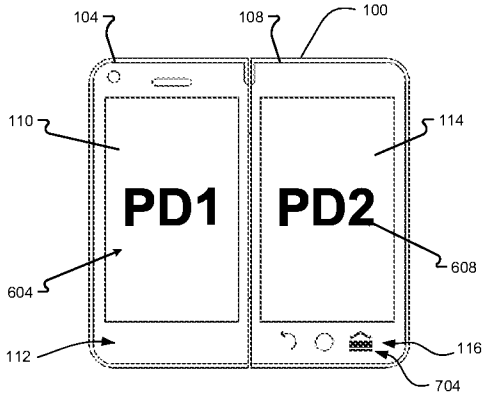


FIG. 7B

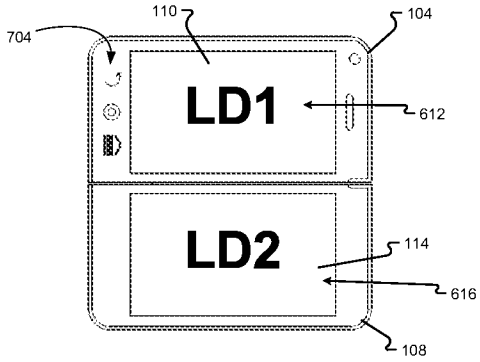


FIG. 8A

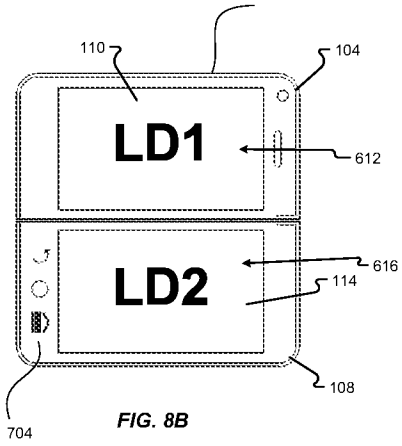


FIG. 8B

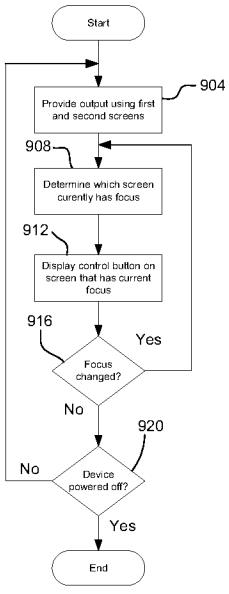


Fig. 9

[0100] The device 100 manages desktops and/or windows in one or more displays, as shown in Figs. 7A through 7B. The user interface 700 or a portion thereof is shown in Fig. 7A. The user interface 700 can include a first touch sensitive display 110, a second touch sensitive display 114, and/or a configurable area 116. A window can be displayed on either or both of the first touch sensitive display 110 and/or the second touch sensitive display 114. A user may provide input into the device 100 using by selecting a button in the configurable area 116. In embodiments, there may be a configurable area associated with each of the touch sensitive displays 110, 114. For example, configurable area 116 may be associated with the second touch sensitive display 114, and a configurable area 112 (shown in Fig. 9A) may be associated with the first touch sensitive display 110. The configurable area 116,112 may be provided only for one of the first touch sensitive display 110 or second touch sensitive display 114 and may not be provided together. In embodiments, which configurable area 116 or 112 is provided is dependent on which window associated with either first touch sensitive display 110 or second touch sensitive display 114 has focus.

[0101] Focus indicates the component of the user interface 700 which is currently selected to receive input. Text entered into a keyboard or information pasted from a clipboard is sent to the component which currently has the focus. Moving the focus away from one window to another window is known as an event associated with both windows. Typically, the focus is withdrawn from a window by giving another window the focus. In embodiments, the device must also determine which configurable area 112,116 should be provided based on the focus and on the device state and/or orientation.

[0102] In embodiments, window 2 712 displayed on the second touch sensitive display 114 of the second screen currently has focus, as shown in Fig. 7A. Thus, configurable area 116 is also provided because configurable area 116 is on the second screen. Focus is shown in Fig. 7A by the dark dashed line around window 2 712. When the device 100 changes state, changes orientation, or due to other events, the focus can change. An focus change based on a display change is described in conjunction with Figs. 7B through Fig. 8.

[0103] An embodiment of a method 800 for changing focus is shown in Fig. 8. While a general order for the steps of the method 800 is shown in Fig. 8. Generally, the method 800 starts with a start operation 804 and ends with an end operation 832. The method 800 can include more or fewer steps or can arrange the order of the steps differently than those shown in

Fig. 8. The method 800 can be executed as a set of computer-executable instructions executed by a computer system and encoded or stored on a computer readable medium. Hereinafter, the method 800 shall be explained with reference to the systems, components, modules, software, data structures, user interfaces, etc. described in conjunction with Figs. 1-7D.

[0104] In embodiments, the device 100 can establish focus on a window in the user interface, in step 808. For example, the device 100 can establish focus on window 2 712 in the user interface 700. In embodiments, the Window Management Module 532 establishes focus based on information provided from the Task Management Module 540. After focus is established, a gesture may be received to reveal the desktop, in step 812. For example, a user may execute a spread gesture 410, select a button 116c, 112c, or conduct some other user interface action to reveal the desktop. As shown in Fig. 7B, the desktop 720 may then be revealed, in step 816. In the embodiment shown in Fig. 7B, window 1 708 and window 2 712 are suspended and appear to the user to slide from the user interface 700, as represented by arrows 716. The sliding action of window 1 708 and window 2 712 appears to reveal the desktop 720 that appears under the moving windows 708,712. The revealed desktop 720 is as shown in Fig. 7C.

[0105] In embodiments, focus is generally only applied to a window. Thus, with the desktop 720 revealed, no window has focus. Window 2 712 had the focus before the desktop 720 was revealed, as shown in Fig. 7A. When the desktop 720 was revealed, window 2 712 lost the focus. As shown in Fig. 7C, no window has focus, and the configurable area 116 remains lit or highlighted. In Fig. 7D, another gesture or user input causes the device 100 to hide the desktop 720, in step 820. In embodiments, the windows 708, 712, which were the last windows displayed, appear to slide over the desktop 720, as represented by arrows 724. Thus, the previously displayed windows 708, 712 are displayed when the desktop 720 is hidden, in step 824.

[0106] When the previously displayed windows 708,712 are again displayed, no window has focus because window 2 712 had lost the focus. Thus, the device 100 reestablishes focus on the window that previously had focus, in step 828. In embodiments, a data structure associated with all or each of the windows may store an indication as to which window last had focus. In the embodiment shown in Fig. 7D, window 2 712 last had focus. The Window Management Module 532 based on information provided from the Task Management Module 540 or other framework 520 component can reestablish window 2 712 with the focus. Thus, rather than moving focus to

the first touch sensitive display 110 on the primary screen 104, the device 100 maintains focus with the last window to have focus. This configuration maintains the display or provision of the configurable area 116 and is more intuitive for the user.

[0107] An embodiment of another method 1000 for changing focus is shown in Fig. 10. While a general order for the steps of the method 1000 is shown in Fig. 10. Generally, the method 1000 starts with a start operation 1004 and ends with an end operation 1028. The method 1000 can include more or fewer steps or can arrange the order of the steps differently than those shown in Fig. 10. The method 1000 can be executed as a set of computer-executable instructions executed by a computer system and encoded or stored on a computer readable medium. Hereinafter, the method 1000 shall be explained with reference to the systems, components, modules, software, data structures, user interfaces, etc. described in conjunction with Figs. 1-9B.

[0108] In embodiments, the device 100 can establish focus on a window in the user interface, in step 1008. For example, the device 100 can establish focus on a full screen window 904 in the user interface 900, as shown in Fig. 9A. The window 904 spans both the first and second displays 110, 114. In embodiments, the Window Management Module 532 establishes focus based on information provided from the Task Management Module 540. In response to the focus establishment, the device 100 may also determine which configurable area 112,116 is provide or configured, in step 1012. In the embodiment shown in Fig. 9A, configurable area 112 is provided as the full screen window 904 covers both displays 110,114 and the first touch sensitive display 110 is on the primary screen 104. Thus, the configurable area 112 of the primary screen 104 is provided in the configuration shown in Fig. 9A.

[0109] Afrer focus is established, an orientation change 908 may be received to change the device from a open portrait state 320 to open landscape state 348, in step 1016. For example, a user may turn the device 100 in the direction of the arrow 908 to change the orientation of the device 100. As shown in Fig. 9B, the device 100 is now in landscape orientation. In embodiments, the Window Management Module 532 maintains focus on window 1 904 based on information provided from the Task Management Module 540, in step 1020. However, the configurable area configuration is changed to provide configurable area 116 instead of configurable area 112 (as represented by arrow 912), in step 1024. The configurable area 116 can be moved in the configuration shown in Fig. 9B to give the user more area to hold the device 100 around the first touch sensitive display 110 at the bottom of the device 100. Thus, the

Window Management Module 532 determines that the configurable area configuration should change based on information provided from the Task Management Module 540, even though the focus did not change from window 1 904.

[0110] An embodiment of another method 1200 for changing focus is shown in Fig. 12. While a general order for the steps of the method 1200 is shown in Fig. 12. Generally, the method 1200 starts with a start operation 1204 and ends with an end operation 1228. The method 1200 can include more or fewer steps or can arrange the order of the steps differently than those shown in Fig. 12. The method 1200 can be executed as a set of computer-executable instructions executed by a computer system and encoded or stored on a computer readable medium. Hereinafrer, the method 1200 shall be explained with reference to the systems, components, modules, software, data structures, user interfaces, etc. described in conjunction with Figs. 1-11B.

[0111] In embodiments, the device 120 can establish focus in the user interface, in step 1208. However, it is possible that focus is not established on a window but appears to be on the desktop. For example, the device 120 may not have set the focus on window 1 1108 but may make it appear that the desktop 1104 has focus by having the configurable area 112 provided rather than configurable area 116. The active window 1 1108 may occupy only the first touch sensitive display 114 and not the first touch sensitive display 110. In embodiments, the Window Management Module 532 establishes focus based on information provided from the Task Management Module 540. In response to the focus establishment, the device 120 may also determine which configurable area 112,116 is provide or configured, in step 1212. In the embodiment shown in Fig. 11A, configurable area 112 is provided as window 1 1108 does not have focus and the first touch sensitive display 112 is on the primary screen 124. Thus, the configurable area 112 of the primary screen 124 is provided in the configuration shown in Fig. 11A.

[0112] Afrer focus is established, an orientation change 1112 may be received to change the device from a open portrait state 320 to open landscape state 348, in step 1216. For example, a user may turn the device 120 in the direction of the arrow 1112 to change the orientation of the device 120. As shown in Fig. 11B, the device 120 is now in landscape orientation. In embodiments, the Window Management Module 532 or some other component may change the display of the active window 1108 by expanding the window 1108 into a full screen window by

displaying an action called a "gravity drop." A gravity drop appears to make the bottom part of the window 1108 drop to the bottom part of touch sensitive display 110. Thus, the active window 1108 becomes a full screen window. After the window 1108 occupies the entire display area, the Window Management Module 532 may change the focus to window 1 1108 based on information provided from the Task Management Module 540, in step 1220.

[0113] However, the configurable area configuration is also changed to provide configurable area 116 instead of configurable area 112 (as represented by arrow 1116), in step 1224. The configurable area 116 can be moved in the configuration shown in Fig. 11B to give the user more area to hold the device 120 around the first touch sensitive display 110 at the bottom of the device 120. Thus, the Window Management Module 532 determines that the configurable area configuration should change based on information provided from the Task Management Module 540, even though the focus did not change from window 1 1108.

[0114] The exemplary systems and methods of this disclosure have been described in relation to a window stack associated with a multi-screen device. However, to avoid unnecessarily obscuring the present disclosure, the preceding description omits a number of known structures and devices. This omission is not to be construed as a limitation of the scopes of the claims. Specific details are set forth to provide an understanding of the present disclosure. It should however be appreciated that the present disclosure may be practiced in a variety of ways beyond the specific detail set forth herein.

[0115] Furthermore, while the exemplary aspects, embodiments, and/or configurations illustrated herein show the various components of the system collocated, certain components of the system can be located remotely, at distant portions of a distributed network, such as a LAN and/or the Internet, or within a dedicated system. Thus, it should be appreciated, that the components of the system can be combined in to one or more devices, such as the software modules of Figs. 5A and 5B, or collocated on a particular node of a distributed network, such as an analog and/or digital telecommunications network, a packet-switch network, or a circuit-switched network. It will be appreciated from the preceding description, and for reasons of computational efficiency, that the components of the system can be arranged at any location within a distributed network of components without affecting the operation of the system. For example, the various components can be located in a switch such as a PBX and media server, gateway, in one or more communications devices, at one or more users' premises, or some

combination thereof. Similarly, one or more functional portions of the system could be distributed between a telecommunications device(s) and an associated computing device.

[0116] Furthermore, it should be appreciated that the various links connecting the elements can be wired or wireless links, or any combination thereof, or any other known or later developed element(s) that is capable of supplying and/or communicating data to and from the connected elements. These wired or wireless links can also be secure links and may be capable of communicating encrypted information. Transmission media used as links, for example, can be any suitable carrier for electrical signals, including coaxial cables, copper wire and fiber optics, and may take the form of acoustic or light waves, such as those generated during radio-wave and infra-red data communications.

[0117] Also, while the flowcharts have been discussed and illustrated in relation to a particular sequence of events, it should be appreciated that changes, additions, and omissions to this sequence can occur without materially affecting the operation of the disclosed embodiments, configuration, and aspects.

[0118] A number of variations and modifications of the disclosure can be used. It would be possible to provide for some features of the disclosure without providing others. For example in one alternative embodiment, the window stack 700 resembles a carousel or rolodex rather than a deck of cards. Thus, windows may circulate from one touch sensitive display 110 to another touch sensitive display 114. For example, window may be push to the right and end up at the end of stack behind another window. If the stack continues to move to the right, eventually the window would appear on the second touch sensitive display even though the window was opened in the first touch sensitive display 110. These movements and changes in the stacks may be managed using the methods and logical data structures discussed above. In another alternative embodiment, there may be other arrangement for the window stack.

[0119] In yet another embodiment, the systems and methods of this disclosure can be implemented in conjunction with a special purpose computer, a programmed microprocessor or microcontroller and peripheral integrated circuit element(s), an ASIC or other integrated circuit, a digital signal processor, a hard-wired electronic or logic circuit such as discrete element circuit, a programmable logic device or gate array such as PLD, PLA, FPGA, PAL, special purpose computer, any comparable means, or the like. In general, any device(s) or means capable of implementing the methodology illustrated herein can be used to implement the various aspects of

this disclosure. Exemplary hardware that can be used for the disclosed embodiments, configurations and aspects includes computers, handheld devices, telephones (e.g., cellular, Internet enabled, digital, analog, hybrids, and others), and other hardware known in the art. Some of these devices include processors (e.g., a single or multiple microprocessors), memory, nonvolatile storage, input devices, and output devices. Furthermore, alternative software implementations including, but not limited to, distributed processing or component/object distributed processing, parallel processing, or virtual machine processing can also be constructed to implement the methods described herein.

[0120] In yet another embodiment, the disclosed methods may be readily implemented in conjunction with software using object or object-oriented software development environments that provide portable source code that can be used on a variety of computer or workstation platforms. Alternatively, the disclosed system may be implemented partially or fully in hardware using standard logic circuits or VLSI design. Whether software or hardware is used to implement the systems in accordance with this disclosure is dependent on the speed and/or efficiency requirements of the system, the particular function, and the particular software or hardware systems or microprocessor or microcomputer systems being utilized.

[0121] In yet another embodiment, the disclosed methods may be partially implemented in software that can be stored on a storage medium, executed on programmed general-purpose computer with the cooperation of a controller and memory, a special purpose computer, a microprocessor, or the like. In these instances, the systems and methods of this disclosure can be implemented as program embedded on personal computer such as an applet, JAVA® or CGI script, as a resource residing on a server or computer workstation, as a routine embedded in a dedicated measurement system, system component, or the like. The system can also be implemented by physically incorporating the system and/or method into a software and/or hardware system.

[0122] Although the present disclosure describes components and functions implemented in the aspects, embodiments, and/or configurations with reference to particular standards and protocols, the aspects, embodiments, and/or configurations are not limited to such standards and protocols. Other similar standards and protocols not mentioned herein are in existence and are considered to be included in the present disclosure. Moreover, the standards and protocols mentioned herein and other similar standards and protocols not mentioned herein are periodically

superseded by faster or more effective equivalents having essentially the same functions. Such replacement standards and protocols having the same functions are considered equivalents included in the present disclosure.

[0123] The present disclosure, in various aspects, embodiments, and/or configurations, includes components, methods, processes, systems and/or apparatus substantially as depicted and described herein, including various aspects, embodiments, configurations embodiments, subcombinations, and/or subsets thereof. Those of skill in the art will understand how to make and use the disclosed aspects, embodiments, and/or configurations afrer understanding the present disclosure. The present disclosure, in various aspects, embodiments, and/or configurations, includes providing devices and processes in the absence of items not depicted and/or described herein or in various aspects, embodiments, and/or configurations hereof, including in the absence of such items as may have been used in previous devices or processes, e.g., for improving performance, achieving ease and\or reducing cost of implementation.

[0124] The foregoing discussion has been presented for purposes of illustration and description. The foregoing is not intended to limit the disclosure to the form or forms disclosed herein. In the foregoing Detailed Description for example, various features of the disclosure are grouped together in one or more aspects, embodiments, and/or configurations for the purpose of streamlining the disclosure. The features of the aspects, embodiments, and/or configurations of the disclosure may be combined in alternate aspects, embodiments, and/or configurations other than those discussed above. This method of disclosure is not to be interpreted as reflecting an intention that the claims require more features than are expressly recited in each claim. Rather, as the following claims reflect, inventive aspects lie in less than all features of a single foregoing disclosed aspect, embodiment, and/or configuration. Thus, the following claims are hereby incorporated into this Detailed Description, with each claim standing on its own as a separate preferred embodiment of the disclosure.

[0125] Moreover, though the description has included description of one or more aspects, embodiments, and/or configurations and certain variations and modifications, other variations, combinations, and modifications are within the scope of the disclosure, e.g., as may be within the skill and knowledge of those in the art, afrer understanding the present disclosure. It is intended to obtain rights which include alternative aspects, embodiments, and/or configurations to the extent permitted, including alternate, interchangeable and/or equivalent structures, functions,

ranges or steps to those claimed, whether or not such alternate, interchangeable and/or equivalent structures, functions, ranges or steps are disclosed herein, and without intending to publicly dedicate any patentable subject matter.

What is claimed is:

1. A computer readable medium, having stored thereon, computer-executable instructions executable by a processor, the computer-executable instructions causing the processor to execute a method for changing focus in a multi-screen device, the computer-executable instructions comprising:

instructions to establish focus on a window;

instructions to determine a first configurable area to display;

instructions to receive an orientation change;

instructions to maintain focus on window; and

instructions to change from the first configurable area to a second configurable area.

2. The computer readable medium as defined in claim 1, wherein the orientation changes the multi-screen from a portrait orientation to a landscape orientation.

3. The computer readable medium as defined in claim 2, wherein the first configurable area is on a first screen and the second configurable area is on a second screen.

4. The computer readable medium as defined in claim 3, wherein the window is a full screen window.

5. The computer readable medium as defined in claim 4, wherein the first screen includes a first touch sensitive display and the second screen includes a second touch sensitive display.

6. The computer readable medium as defined in claim 5, wherein the full screen window is displayed on both the first touch sensitive display and the second touch sensitive display.

7. The computer readable medium as defined in claim 6, wherein, in landscape orientation, the first screen is at a bottom of the multi-screen device and the second screen is at a top of the multi-screen device.

8. The computer readable medium as defined in claim 7, wherein the multi-screen device is reconfigured to move the configurable area from the bottom of the multi-screen device to the top of the multi-screen device.

9. A device comprising:

a first screen comprising:

a first touch sensitive display;

a first configurable area;

a second screen comprising:

a second touch sensitive display;

a second configurable area;

wherein a first window is a full screen window and is displayed across the first and the second touch sensitive display;

a memory;

a processor in communication with the memory and each of the at least two screens, the processor operable to:

establish focus on the first window;

determine to provide the first configurable area;

receive an orientation change, wherein the orientation change changes the device from a portrait orientation to a landscape orientation;

maintain focus on the first window; and

change from the first configurable area to a second configurable area.

10. The device as defined in claim 9, wherein the first screen is a primary screen.

11. The device as defined in claim 10, wherein the first screen is on a lefr of the device in the portrait orientation and at a bottom of the device in the landscape orientation.

12. The device as defined in claim 11, wherein device changes the screen onto which to provide the configurable area.

13. The device as defined in claim 12, wherein, when the second configurable area is at the top of the device in the landscape orientation.

14. The device as defined in claim 13, wherein an orientation of the first window changes from portrait to landscape in response to the orientation change of the device.

15. The device as defined in claim 14, wherein at least one input into the configurable area affect the first window.

16. A method for presenting a display for a multi-screen device, the method comprising:

establishing focus on a first window while the multi-screen device is in a portrait orientation, wherein the first window is a full screen window;

determining to provide a first configurable area;

receiving an orientation change, wherein the orientation change changes the device from the portrait orientation to a landscape orientation;

maintaining focus on the first window; and

changing from the first configurable area to a second configurable area.

17. The method defined in claim 16, wherein the multi-screen device comprises:

a first screen comprising:

a first touch sensitive display;

the first configurable area;

a second screen comprising:

a second touch sensitive display;

a second configurable area.

18. The method defined in claim 17, wherein the first screen is on a left of the device in the portrait orientation and at a bottom of the device in the landscape orientation.

19. The method defined in claim 18, wherein, when the second configurable area is at the top of the device in the landscape orientation.

20. The method defined in claim 18, wherein the first screen is a primary screen.

ABSTRACT

Systems and methods are provides for adjusting focus during an orientation change. A full screen window has focus before the orientation change. After the device is oriented in the landscape orientation, the focus is maintained on the window. However, a configurable area associated with at least one screen that displays the full screen window is changed to display the configurable area on the screen that is at the top of the device.

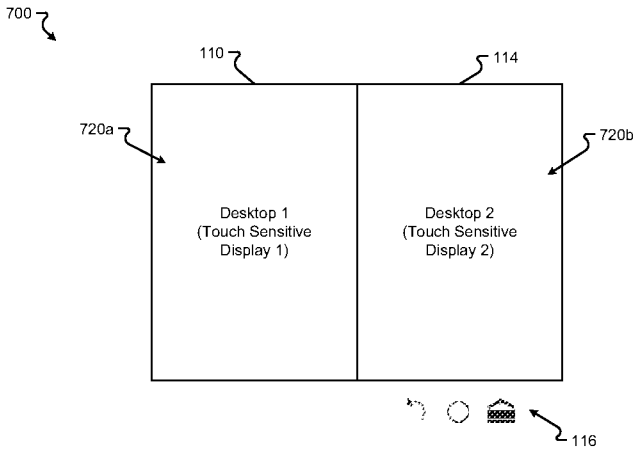


FIG. 7C

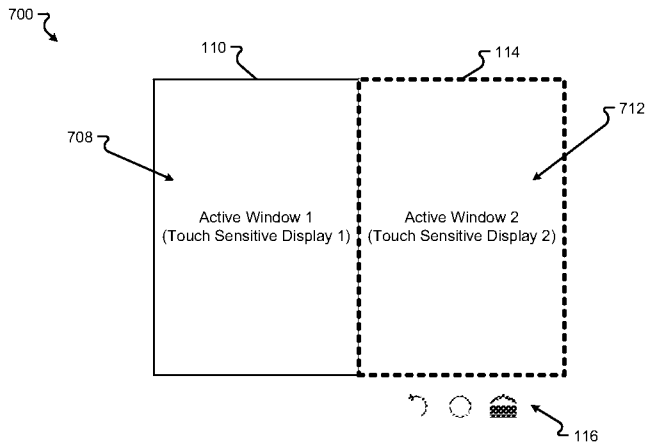


FIG. 7A

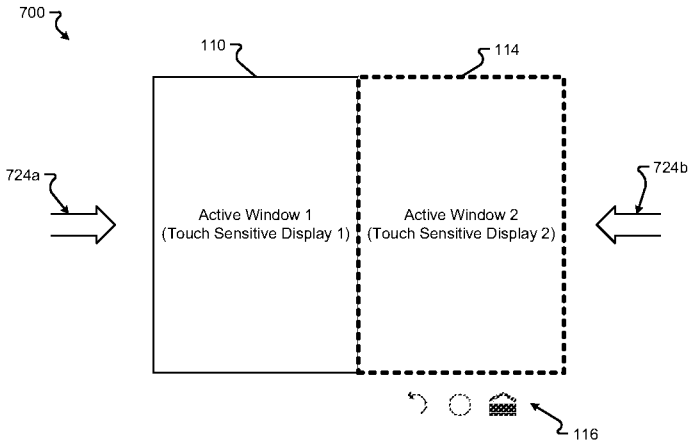


FIG. 7D

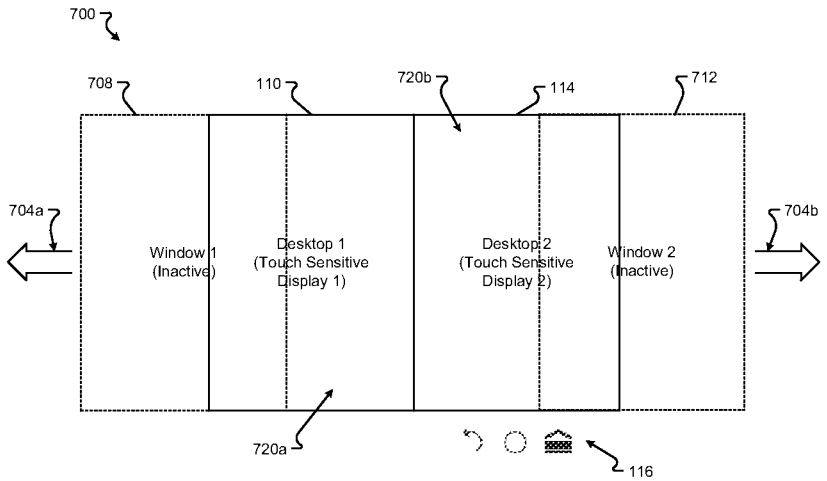


FIG. 7B

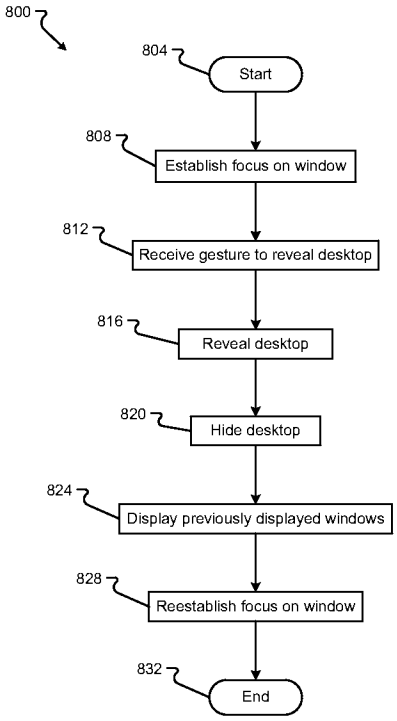


FIG. 8

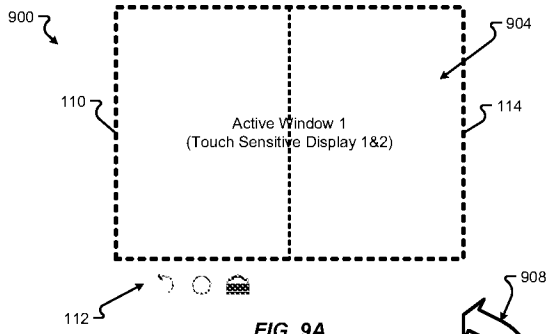


FIG. 9A

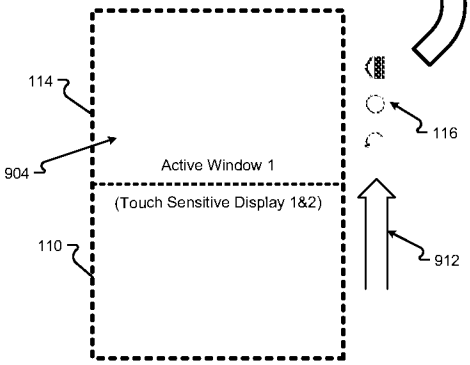


FIG. 9B

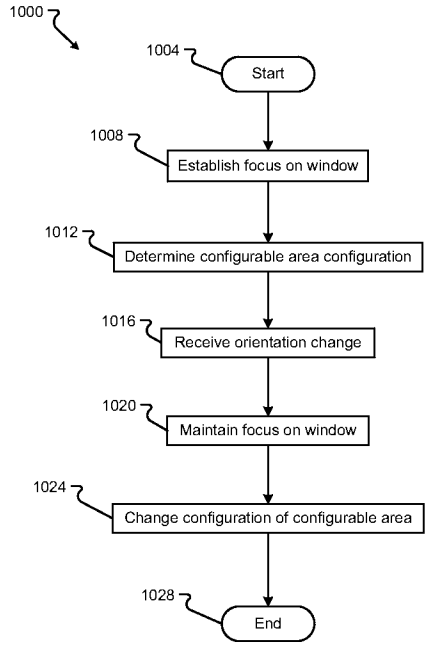


FIG. 10

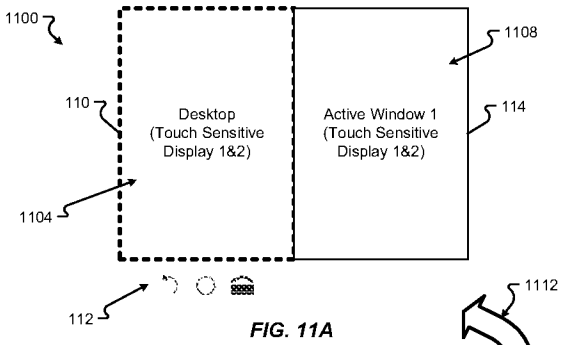


FIG. 11A

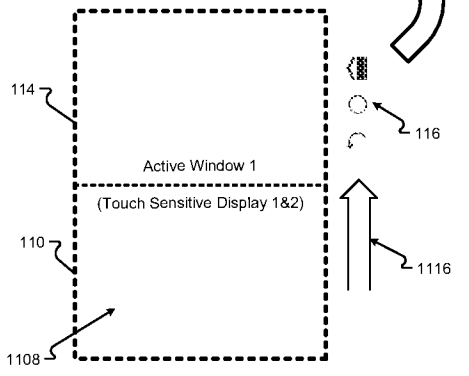


FIG. 11B

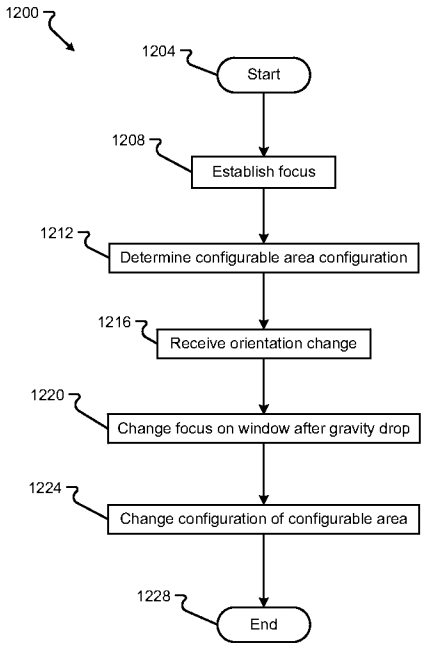

1200
1204
Start
1208
Establish focus
1212
Determine configurable area configuration
1216
Receive orientation change
1220
Change focus on window after gravity drop
1224
Change configuration of configurable area
1228
End

FIG. 12

[0100] The device 100 manages desktops and/or windows in one or more displays, as shown in Figs. 7A through 7B. The user interface 700 or a portion thereof is shown in Fig. 7A. The user interface 700 can include a first touch sensitive display 110, a second touch sensitive display 114, and/or a configurable area 116. A window can be displayed on either or both of the first touch sensitive display 110 and/or the second touch sensitive display 114. A user may provide input into the device 100 using by selecting a button in the configurable area 116. In embodiments, there may be a configurable area associated with each of the touch sensitive displays 110, 114. For example, configurable area 116 may be associated with the second touch sensitive display 114, and a configurable area 112 (shown in Fig. 9A) may be associated with the first touch sensitive display 110. The configurable area 116,112 may be provided only for one of the first touch sensitive display 110 or second touch sensitive display 114 and may not be provided together. In embodiments, which configurable area 116 or 112 is provided is dependent on which window associated with either first touch sensitive display 110 or second touch sensitive display 114 has focus.

[0101] Focus indicates the component of the user interface 700 which is currently selected to receive input. Text entered into a keyboard or information pasted from a clipboard is sent to the component which currently has the focus. Moving the focus away from one window to another window is known as an event associated with both windows. Typically, the focus is withdrawn from a window by giving another window the focus. In embodiments, the device must also determine which configurable area 112,116 should be provided based on the focus and on the device state and/or orientation.

[0102] In embodiments, window 2 712 displayed on the second touch sensitive display 114 of the second screen currently has focus, as shown in Fig. 7A. Thus, configurable area 116 is also provided because configurable area 116 is on the second screen. Focus is shown in Fig. 7A by the dark dashed line around window 2 712. When the device 100 changes state, changes orientation, or due to other events, the focus can change. An focus change based on a display change is described in conjunction with Figs. 7B through Fig. 8.

[0103] An embodiment of a method 800 for changing focus is shown in Fig. 8. While a general order for the steps of the method 800 is shown in Fig. 8. Generally, the method 800 starts with a start operation 804 and ends with an end operation 832. The method 800 can include more or fewer steps or can arrange the order of the steps differently than those shown in

Fig. 8. The method 800 can be executed as a set of computer-executable instructions executed by a computer system and encoded or stored on a computer readable medium. Hereinafter, the method 800 shall be explained with reference to the systems, components, modules, software, data structures, user interfaces, etc. described in conjunction with Figs. 1-7D.

[0104] In embodiments, the device 100 can establish focus on a window in the user interface, in step 808. For example, the device 100 can establish focus on window 2 712 in the user interface 700. In embodiments, the Window Management Module 532 establishes focus based on information provided from the Task Management Module 540. After focus is established, a gesture may be received to reveal the desktop, in step 812. For example, a user may execute a spread gesture 410, select a button 116c, 112c, or conduct some other user interface action to reveal the desktop. As shown in Fig. 7B, the desktop 720 may then be revealed, in step 816. In the embodiment shown in Fig. 7B, window 1 708 and window 2 712 are suspended and appear to the user to slide from the user interface 700, as represented by arrows 716. The sliding action of window 1 708 and window 2 712 appears to reveal the desktop 720 that appears under the moving windows 708,712. The revealed desktop 720 is as shown in Fig. 7C.

[0105] In embodiments, focus is generally only applied to a window. Thus, with the desktop 720 revealed, no window has focus. Window 2 712 had the focus before the desktop 720 was revealed, as shown in Fig. 7A. When the desktop 720 was revealed, window 2 712 lost the focus. As shown in Fig. 7C, no window has focus, and the configurable area 116 remains lit or highlighted. In Fig. 7D, another gesture or user input causes the device 100 to hide the desktop 720, in step 820. In embodiments, the windows 708, 712, which were the last windows displayed, appear to slide over the desktop 720, as represented by arrows 724. Thus, the previously displayed windows 708, 712 are displayed when the desktop 720 is hidden, in step 824.

[0106] When the previously displayed windows 708,712 are again displayed, no window has focus because window 2 712 had lost the focus. Thus, the device 100 reestablishes focus on the window that previously had focus, in step 828. In embodiments, a data structure associated with all or each of the windows may store an indication as to which window last had focus. In the embodiment shown in Fig. 7D, window 2 712 last had focus. The Window Management Module 532 based on information provided from the Task Management Module 540 or other framework 520 component can reestablish window 2 712 with the focus. Thus, rather than moving focus to

the first touch sensitive display 110 on the primary screen 104, the device 100 maintains focus with the last window to have focus. This configuration maintains the display or provision of the configurable area 116 and is more intuitive for the user.

[0107] An embodiment of another method 1000 for changing focus is shown in Fig. 10. While a general order for the steps of the method 1000 is shown in Fig. 10. Generally, the method 1000 starts with a start operation 1004 and ends with an end operation 1028. The method 1000 can include more or fewer steps or can arrange the order of the steps differently than those shown in Fig. 10. The method 1000 can be executed as a set of computer-executable instructions executed by a computer system and encoded or stored on a computer readable medium. Hereinafter, the method 1000 shall be explained with reference to the systems, components, modules, software, data structures, user interfaces, etc. described in conjunction with Figs. 1-9B.

[0108] In embodiments, the device 100 can establish focus on a window in the user interface, in step 1008. For example, the device 100 can establish focus on a full screen window 904 in the user interface 900, as shown in Fig. 9A. The window 904 spans both the first and second displays 110, 114. In embodiments, the Window Management Module 532 establishes focus based on information provided from the Task Management Module 540. In response to the focus establishment, the device 100 may also determine which configurable area 112,116 is provide or configured, in step 1012. In the embodiment shown in Fig. 9A, configurable area 112 is provided as the full screen window 904 covers both displays 110,114 and the first touch sensitive display 110 is on the primary screen 104. Thus, the configurable area 112 of the primary screen 104 is provided in the configuration shown in Fig. 9A.

[0109] Afrer focus is established, an orientation change 908 may be received to change the device from a open portrait state 320 to open landscape state 348, in step 1016. For example, a user may turn the device 100 in the direction of the arrow 908 to change the orientation of the device 100. As shown in Fig. 9B, the device 100 is now in landscape orientation. In embodiments, the Window Management Module 532 maintains focus on window 1 904 based on information provided from the Task Management Module 540, in step 1020. However, the configurable area configuration is changed to provide configurable area 116 instead of configurable area 112 (as represented by arrow 912), in step 1024. The configurable area 116 can be moved in the configuration shown in Fig. 9B to give the user more area to hold the device 100 around the first touch sensitive display 110 at the bottom of the device 100. Thus, the

Window Management Module 532 determines that the configurable area configuration should change based on information provided from the Task Management Module 540, even though the focus did not change from window 1 904.

[0110] An embodiment of another method 1200 for changing focus is shown in Fig. 12. While a general order for the steps of the method 1200 is shown in Fig. 12. Generally, the method 1200 starts with a start operation 1204 and ends with an end operation 1228. The method 1200 can include more or fewer steps or can arrange the order of the steps differently than those shown in Fig. 12. The method 1200 can be executed as a set of computer-executable instructions executed by a computer system and encoded or stored on a computer readable medium. Hereinafrer, the method 1200 shall be explained with reference to the systems, components, modules, software, data structures, user interfaces, etc. described in conjunction with Figs. 1-11B.

[0111] In embodiments, the device 120 can establish focus in the user interface, in step 1208. However, it is possible that focus is not established on a window but appears to be on the desktop. For example, the device 120 may not have set the focus on window 1 1108 but may make it appear that the desktop 1104 has focus by having the configurable area 112 provided rather than configurable area 116. The active window 1 1108 may occupy only the first touch sensitive display 114 and not the first touch sensitive display 110. In embodiments, the Window Management Module 532 establishes focus based on information provided from the Task Management Module 540. In response to the focus establishment, the device 120 may also determine which configurable area 112,116 is provide or configured, in step 1212. In the embodiment shown in Fig. 11A, configurable area 112 is provided as window 1 1108 does not have focus and the first touch sensitive display 112 is on the primary screen 124. Thus, the configurable area 112 of the primary screen 124 is provided in the configuration shown in Fig. 11A.

[0112] Afrer focus is established, an orientation change 1112 may be received to change the device from a open portrait state 320 to open landscape state 348, in step 1216. For example, a user may turn the device 120 in the direction of the arrow 1112 to change the orientation of the device 120. As shown in Fig. 11B, the device 120 is now in landscape orientation. In embodiments, the Window Management Module 532 or some other component may change the display of the active window 1108 by expanding the window 1108 into a full screen window by

displaying an action called a "gravity drop." A gravity drop appears to make the bottom part of the window 1108 drop to the bottom part of touch sensitive display 110. Thus, the active window 1108 becomes a full screen window. After the window 1108 occupies the entire display area, the Window Management Module 532 may change the focus to window 1 1108 based on information provided from the Task Management Module 540, in step 1220.

[0113] However, the configurable area configuration is also changed to provide configurable area 116 instead of configurable area 112 (as represented by arrow 1116), in step 1224. The configurable area 116 can be moved in the configuration shown in Fig. 11B to give the user more area to hold the device 120 around the first touch sensitive display 110 at the bottom of the device 120. Thus, the Window Management Module 532 determines that the configurable area configuration should change based on information provided from the Task Management Module 540, even though the focus did not change from window 1 1108.

[0114] The exemplary systems and methods of this disclosure have been described in relation to a window stack associated with a multi-screen device. However, to avoid unnecessarily obscuring the present disclosure, the preceding description omits a number of known structures and devices. This omission is not to be construed as a limitation of the scopes of the claims. Specific details are set forth to provide an understanding of the present disclosure. It should however be appreciated that the present disclosure may be practiced in a variety of ways beyond the specific detail set forth herein.

[0115] Furthermore, while the exemplary aspects, embodiments, and/or configurations illustrated herein show the various components of the system collocated, certain components of the system can be located remotely, at distant portions of a distributed network, such as a LAN and/or the Internet, or within a dedicated system. Thus, it should be appreciated, that the components of the system can be combined in to one or more devices, such as the software modules of Figs. 5A and 5B, or collocated on a particular node of a distributed network, such as an analog and/or digital telecommunications network, a packet-switch network, or a circuit-switched network. It will be appreciated from the preceding description, and for reasons of computational efficiency, that the components of the system can be arranged at any location within a distributed network of components without affecting the operation of the system. For example, the various components can be located in a switch such as a PBX and media server, gateway, in one or more communications devices, at one or more users' premises, or some

combination thereof. Similarly, one or more functional portions of the system could be distributed between a telecommunications device(s) and an associated computing device.

[0116] Furthermore, it should be appreciated that the various links connecting the elements can be wired or wireless links, or any combination thereof, or any other known or later developed element(s) that is capable of supplying and/or communicating data to and from the connected elements. These wired or wireless links can also be secure links and may be capable of communicating encrypted information. Transmission media used as links, for example, can be any suitable carrier for electrical signals, including coaxial cables, copper wire and fiber optics, and may take the form of acoustic or light waves, such as those generated during radio-wave and infra-red data communications.

[0117] Also, while the flowcharts have been discussed and illustrated in relation to a particular sequence of events, it should be appreciated that changes, additions, and omissions to this sequence can occur without materially affecting the operation of the disclosed embodiments, configuration, and aspects.

[0118] A number of variations and modifications of the disclosure can be used. It would be possible to provide for some features of the disclosure without providing others. For example in one alternative embodiment, the window stack 700 resembles a carousel or rolodex rather than a deck of cards. Thus, windows may circulate from one touch sensitive display 110 to another touch sensitive display 114. For example, window may be push to the right and end up at the end of stack behind another window. If the stack continues to move to the right, eventually the window would appear on the second touch sensitive display even though the window was opened in the first touch sensitive display 110. These movements and changes in the stacks may be managed using the methods and logical data structures discussed above. In another alternative embodiment, there may be other arrangement for the window stack.

[0119] In yet another embodiment, the systems and methods of this disclosure can be implemented in conjunction with a special purpose computer, a programmed microprocessor or microcontroller and peripheral integrated circuit element(s), an ASIC or other integrated circuit, a digital signal processor, a hard-wired electronic or logic circuit such as discrete element circuit, a programmable logic device or gate array such as PLD, PLA, FPGA, PAL, special purpose computer, any comparable means, or the like. In general, any device(s) or means capable of implementing the methodology illustrated herein can be used to implement the various aspects of

this disclosure. Exemplary hardware that can be used for the disclosed embodiments, configurations and aspects includes computers, handheld devices, telephones (e.g., cellular, Internet enabled, digital, analog, hybrids, and others), and other hardware known in the art. Some of these devices include processors (e.g., a single or multiple microprocessors), memory, nonvolatile storage, input devices, and output devices. Furthermore, alternative software implementations including, but not limited to, distributed processing or component/object distributed processing, parallel processing, or virtual machine processing can also be constructed to implement the methods described herein.

[0120] In yet another embodiment, the disclosed methods may be readily implemented in conjunction with software using object or object-oriented software development environments that provide portable source code that can be used on a variety of computer or workstation platforms. Alternatively, the disclosed system may be implemented partially or fully in hardware using standard logic circuits or VLSI design. Whether software or hardware is used to implement the systems in accordance with this disclosure is dependent on the speed and/or efficiency requirements of the system, the particular function, and the particular software or hardware systems or microprocessor or microcomputer systems being utilized.

[0121] In yet another embodiment, the disclosed methods may be partially implemented in software that can be stored on a storage medium, executed on programmed general-purpose computer with the cooperation of a controller and memory, a special purpose computer, a microprocessor, or the like. In these instances, the systems and methods of this disclosure can be implemented as program embedded on personal computer such as an applet, JAVA® or CGI script, as a resource residing on a server or computer workstation, as a routine embedded in a dedicated measurement system, system component, or the like. The system can also be implemented by physically incorporating the system and/or method into a software and/or hardware system.

[0122] Although the present disclosure describes components and functions implemented in the aspects, embodiments, and/or configurations with reference to particular standards and protocols, the aspects, embodiments, and/or configurations are not limited to such standards and protocols. Other similar standards and protocols not mentioned herein are in existence and are considered to be included in the present disclosure. Moreover, the standards and protocols mentioned herein and other similar standards and protocols not mentioned herein are periodically

superseded by faster or more effective equivalents having essentially the same functions. Such replacement standards and protocols having the same functions are considered equivalents included in the present disclosure.

[0123] The present disclosure, in various aspects, embodiments, and/or configurations, includes components, methods, processes, systems and/or apparatus substantially as depicted and described herein, including various aspects, embodiments, configurations embodiments, subcombinations, and/or subsets thereof. Those of skill in the art will understand how to make and use the disclosed aspects, embodiments, and/or configurations afrer understanding the present disclosure. The present disclosure, in various aspects, embodiments, and/or configurations, includes providing devices and processes in the absence of items not depicted and/or described herein or in various aspects, embodiments, and/or configurations hereof, including in the absence of such items as may have been used in previous devices or processes, e.g., for improving performance, achieving ease and\or reducing cost of implementation.

[0124] The foregoing discussion has been presented for purposes of illustration and description. The foregoing is not intended to limit the disclosure to the form or forms disclosed herein. In the foregoing Detailed Description for example, various features of the disclosure are grouped together in one or more aspects, embodiments, and/or configurations for the purpose of streamlining the disclosure. The features of the aspects, embodiments, and/or configurations of the disclosure may be combined in alternate aspects, embodiments, and/or configurations other than those discussed above. This method of disclosure is not to be interpreted as reflecting an intention that the claims require more features than are expressly recited in each claim. Rather, as the following claims reflect, inventive aspects lie in less than all features of a single foregoing disclosed aspect, embodiment, and/or configuration. Thus, the following claims are hereby incorporated into this Detailed Description, with each claim standing on its own as a separate preferred embodiment of the disclosure.

[0125] Moreover, though the description has included description of one or more aspects, embodiments, and/or configurations and certain variations and modifications, other variations, combinations, and modifications are within the scope of the disclosure, e.g., as may be within the skill and knowledge of those in the art, afrer understanding the present disclosure. It is intended to obtain rights which include alternative aspects, embodiments, and/or configurations to the extent permitted, including alternate, interchangeable and/or equivalent structures, functions,

ranges or steps to those claimed, whether or not such alternate, interchangeable and/or equivalent structures, functions, ranges or steps are disclosed herein, and without intending to publicly dedicate any patentable subject matter.

What is claimed is:

1. A computer readable medium, having stored thereon, computer-executable instructions executable by a processor, the computer-executable instructions causing the processor to execute a method for changing focus in a multi-screen device, the computer-executable instructions comprising:

instructions to determine a first configurable area to display;

instructions to receive an orientation change;

instructions to expand the window into a full screen window;

instructions to change focus to the window; and

instructions to change from the first configurable area to a second configurable area.

2. The computer readable medium as defined in claim 1, wherein the orientation changes the multi-screen from a portrait orientation to a landscape orientation.

3. The computer readable medium as defined in claim 2, wherein the first configurable area is on a first screen and the second configurable area is on a second screen.

4. The computer readable medium as defined in claim 3, wherein the window is a displayed on one of two or more displays before the orientation change.

5. The computer readable medium as defined in claim 4, wherein the first screen includes a first touch sensitive display and the second screen includes a second touch sensitive display.

6. The computer readable medium as defined in claim 5, wherein the full screen window is displayed on both the first touch sensitive display and the second touch sensitive display.

7. The computer readable medium as defined in claim 6, wherein, in landscape orientation, the first screen is at a bottom of the multi-screen device and the second screen is at a top of the multi-screen device.

8. The computer readable medium as defined in claim 7, wherein the multi-screen device is reconfigured to move the configurable area from the bottom of the multi-screen device to the top of the multi-screen device.

9. A device comprising:

a first screen comprising:

a first touch sensitive display;

a first configurable area;

a second screen comprising:

a second touch sensitive display;

a second configurable area;

wherein a first window is displayed on the second touch sensitive display and does not have focus;

a memory;

a processor in communication with the memory and each of the at least two screens, the processor operable to:

determine to provide the first configurable area;

receive an orientation change, wherein the orientation change changes the device from a portrait orientation to a landscape orientation;

expand the first window into a full screen window displayed on both the first touch sensitive display and the second touch sensitive display;

change focus to the first window; and

change from the first configurable area to a second configurable area.

10. The device as defined in claim 9, wherein the first screen is a primary screen.

11. The device as defined in claim 10, wherein the first screen is on a lefr of the device in the portrait orientation and at a bottom of the device in the landscape orientation.

12. The device as defined in claim 11, wherein the device changes the screen onto which to provide the configurable area.

13. The device as defined in claim 12, wherein, when the second configurable area is at the top of the device in the landscape orientation.

14. The device as defined in claim 13, wherein an orientation of the first window changes from portrait to landscape in response to the orientation change of the device.

15. The device as defined in claim 14, wherein at least one input into the second configurable area affects the first window.

16. A method for presenting a display for a multi-screen device, the method comprising:

displaying a first window on a second touch sensitive display;

determining to provide a first configurable area, wherein the first configurable area is not associated with the second touch sensitive display;

receiving an orientation change, wherein the orientation change changes the device ftom the portrait orientation to a landscape orientation;

expanding the first window, by a gravity drop, to by displayed on the second touch sensitive display and a first touch sensitive display;

changing focus to the first window; and

changing ftom the first configurable area to a second configurable area.

17. The method defined in claim 16, wherein the multi-screen device comprises:

a first screen comprising:

the first touch sensitive display;

the first configurable area;

a second screen comprising:

the second touch sensitive display;

the second configurable area.

18. The method defined in claim 17, wherein the first screen is on a left of the device in the portrait orientation and at a bottom of the device in the landscape orientation.

19. The method defined in claim 18, wherein the second configurable area is at the top of the device in the landscape orientation.

20. The method defined in claim 18, wherein the first screen is a primary screen.

ABSTRACT

Systems and methods are provides for adjusting focus during an orientation change. A window is displayed on a single display before the orientation change. After the device is oriented in the landscape orientation, the window expands into a full screen window being displayed on two displays. Further, the focus is changed to the window. And, a configurable area associated with at least one screen that displays the full screen window is changed to display the configurable area on the screen that is at the top of the device.

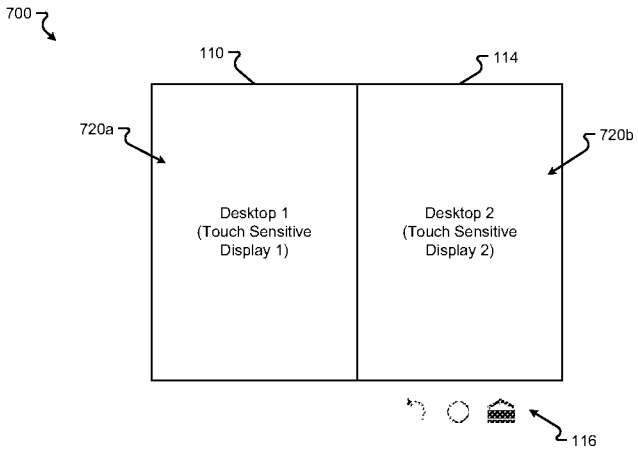


FIG. 7C

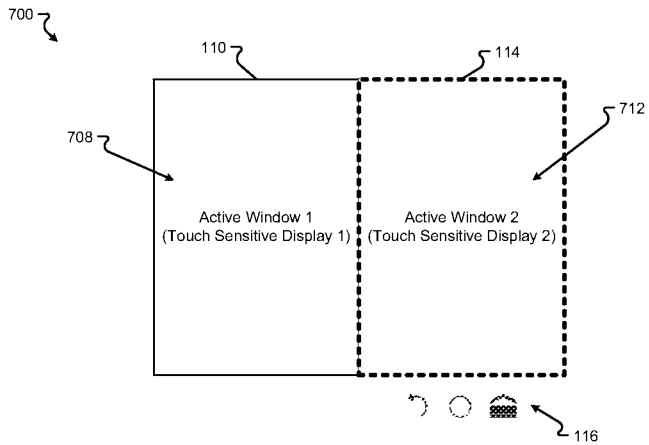


FIG. 7A

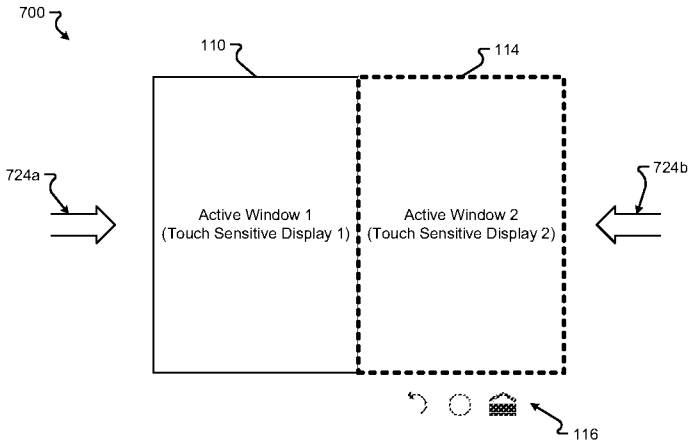


FIG. 7D

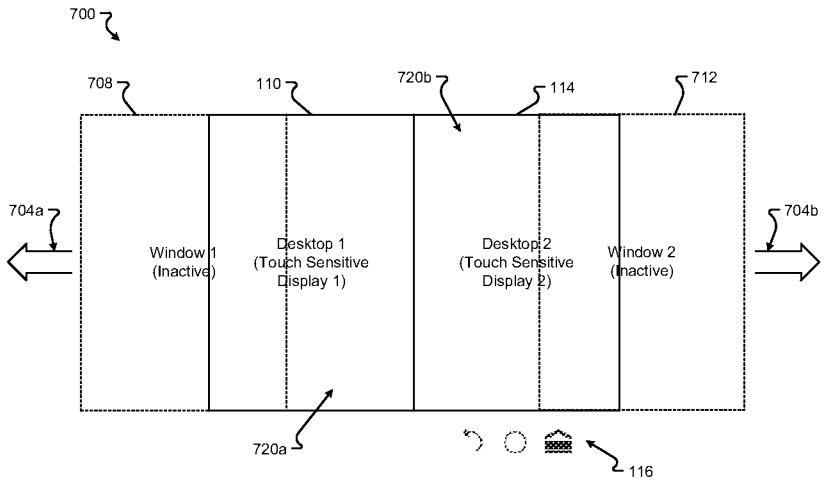


FIG. 7B

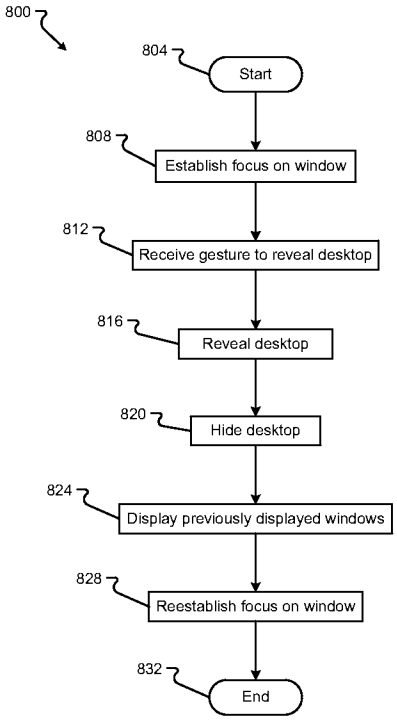


FIG. 8

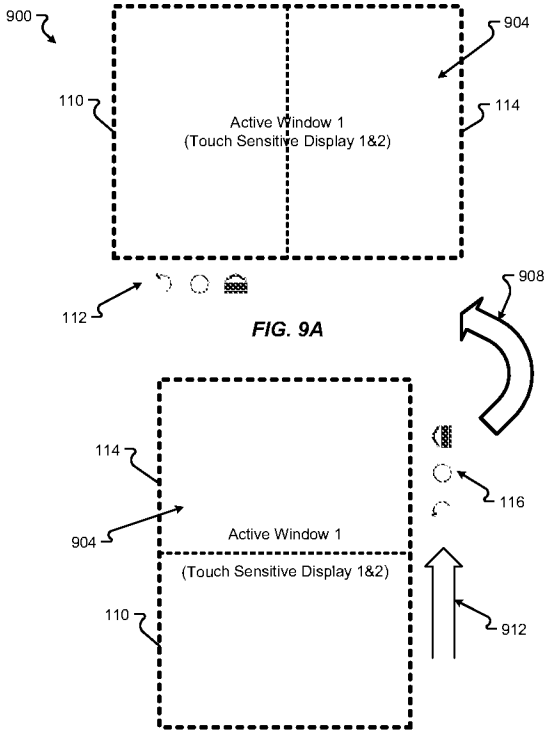


FIG. 9A

FIG. 9B

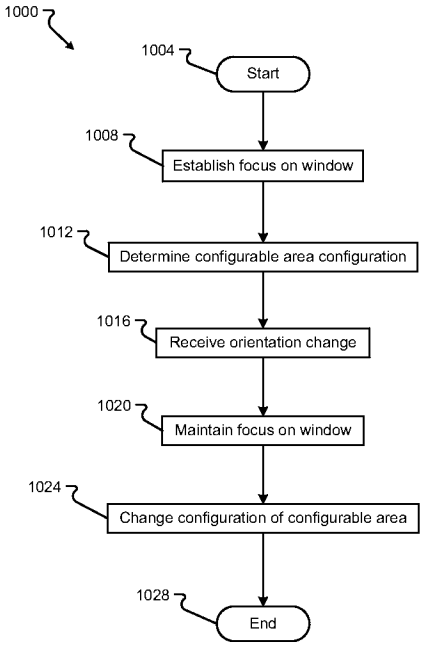


FIG. 10

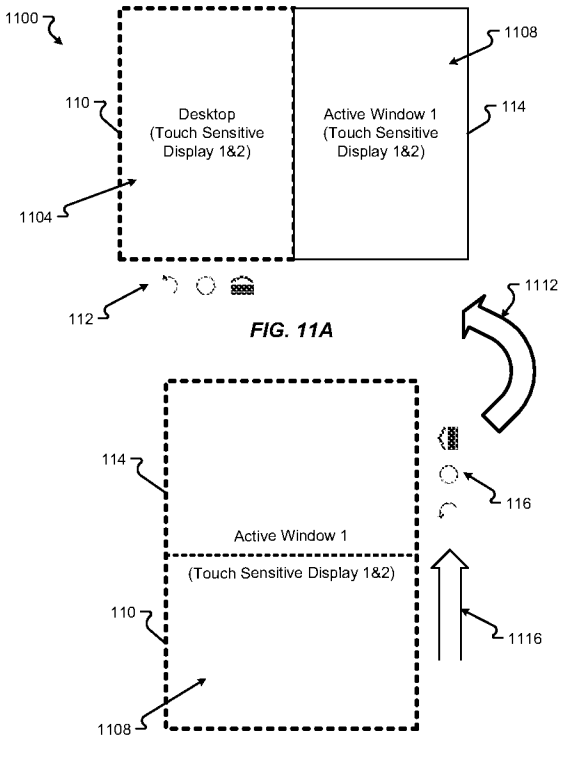


FIG. 11A

FIG. 11B

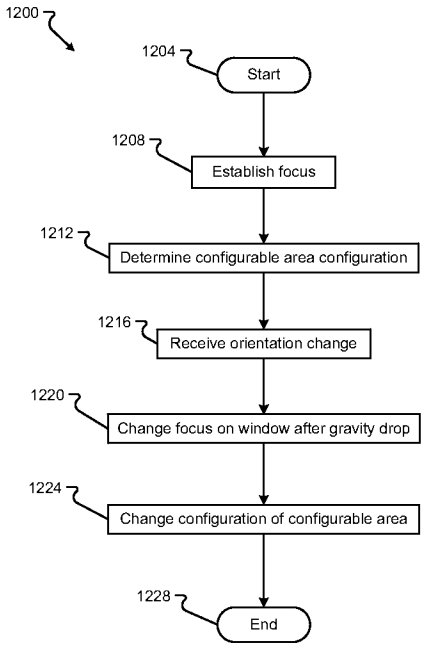

1200
1204
Start
1208
Establish focus
1212
Determine configurable area configuration
1216
Receive orientation change
1220
Change focus on window after gravity drop
1224
Change configuration of configurable area
1228
End

FIG. 12

[0100] The device 100 manages desktops and/or windows in one or more displays, as shown in Figs. 7A through 7B. The user interface 700 or a portion thereof is shown in Fig. 7A. The user interface 700 can include a first touch sensitive display 110, a second touch sensitive display 114, and/or a configurable area 116. A window can be displayed on either or both of the first touch sensitive display 110 and/or the second touch sensitive display 114. A user may provide input into the device 100 using by selecting a button in the configurable area 116. In embodiments, there may be a configurable area associated with each of the touch sensitive displays 110, 114. For example, configurable area 116 may be associated with the second touch sensitive display 114, and a configurable area 112 (shown in Fig. 9A) may be associated with the first touch sensitive display 110. The configurable area 116,112 may be provided only for one of the first touch sensitive display 110 or second touch sensitive display 114 and may not be provided together. In embodiments, which configurable area 116 or 112 is provided is dependent on which window associated with either first touch sensitive display 110 or second touch sensitive display 114 has focus.

[0101] Focus indicates the component of the user interface 700 which is currently selected to receive input. Text entered into a keyboard or information pasted from a clipboard is sent to the component which currently has the focus. Moving the focus away from one window to another window is known as an event associated with both windows. Typically, the focus is withdrawn from a window by giving another window the focus. In embodiments, the device must also determine which configurable area 112,116 should be provided based on the focus and on the device state and/or orientation.

[0102] In embodiments, window 2 712 displayed on the second touch sensitive display 114 of the second screen currently has focus, as shown in Fig. 7A. Thus, configurable area 116 is also provided because configurable area 116 is on the second screen. Focus is shown in Fig. 7A by the dark dashed line around window 2 712. When the device 100 changes state, changes orientation, or due to other events, the focus can change. An focus change based on a display change is described in conjunction with Figs. 7B through Fig. 8.

[0103] An embodiment of a method 800 for changing focus is shown in Fig. 8. While a general order for the steps of the method 800 is shown in Fig. 8. Generally, the method 800 starts with a start operation 804 and ends with an end operation 832. The method 800 can include more or fewer steps or can arrange the order of the steps differently than those shown in

Fig. 8. The method 800 can be executed as a set of computer-executable instructions executed by a computer system and encoded or stored on a computer readable medium. Hereinafter, the method 800 shall be explained with reference to the systems, components, modules, software, data structures, user interfaces, etc. described in conjunction with Figs. 1-7D.

[0104] In embodiments, the device 100 can establish focus on a window in the user interface, in step 808. For example, the device 100 can establish focus on window 2 712 in the user interface 700. In embodiments, the Window Management Module 532 establishes focus based on information provided from the Task Management Module 540. After focus is established, a gesture may be received to reveal the desktop, in step 812. For example, a user may execute a spread gesture 410, select a button 116c, 112c, or conduct some other user interface action to reveal the desktop. As shown in Fig. 7B, the desktop 720 may then be revealed, in step 816. In the embodiment shown in Fig. 7B, window 1 708 and window 2 712 are suspended and appear to the user to slide from the user interface 700, as represented by arrows 716. The sliding action of window 1 708 and window 2 712 appears to reveal the desktop 720 that appears under the moving windows 708,712. The revealed desktop 720 is as shown in Fig. 7C.

[0105] In embodiments, focus is generally only applied to a window. Thus, with the desktop 720 revealed, no window has focus. Window 2 712 had the focus before the desktop 720 was revealed, as shown in Fig. 7A. When the desktop 720 was revealed, window 2 712 lost the focus. As shown in Fig. 7C, no window has focus, and the configurable area 116 remains lit or highlighted. In Fig. 7D, another gesture or user input causes the device 100 to hide the desktop 720, in step 820. In embodiments, the windows 708, 712, which were the last windows displayed, appear to slide over the desktop 720, as represented by arrows 724. Thus, the previously displayed windows 708, 712 are displayed when the desktop 720 is hidden, in step 824.

[0106] When the previously displayed windows 708,712 are again displayed, no window has focus because window 2 712 had lost the focus. Thus, the device 100 reestablishes focus on the window that previously had focus, in step 828. In embodiments, a data structure associated with all or each of the windows may store an indication as to which window last had focus. In the embodiment shown in Fig. 7D, window 2 712 last had focus. The Window Management Module 532 based on information provided from the Task Management Module 540 or other framework 520 component can reestablish window 2 712 with the focus. Thus, rather than moving focus to

the first touch sensitive display 110 on the primary screen 104, the device 100 maintains focus with the last window to have focus. This configuration maintains the display or provision of the configurable area 116 and is more intuitive for the user.

[0107] An embodiment of another method 1000 for changing focus is shown in Fig. 10. While a general order for the steps of the method 1000 is shown in Fig. 10. Generally, the method 1000 starts with a start operation 1004 and ends with an end operation 1028. The method 1000 can include more or fewer steps or can arrange the order of the steps differently than those shown in Fig. 10. The method 1000 can be executed as a set of computer-executable instructions executed by a computer system and encoded or stored on a computer readable medium. Hereinafter, the method 1000 shall be explained with reference to the systems, components, modules, software, data structures, user interfaces, etc. described in conjunction with Figs. 1-9B.

[0108] In embodiments, the device 100 can establish focus on a window in the user interface, in step 1008. For example, the device 100 can establish focus on a full screen window 904 in the user interface 900, as shown in Fig. 9A. The window 904 spans both the first and second displays 110, 114. In embodiments, the Window Management Module 532 establishes focus based on information provided from the Task Management Module 540. In response to the focus establishment, the device 100 may also determine which configurable area 112,116 is provide or configured, in step 1012. In the embodiment shown in Fig. 9A, configurable area 112 is provided as the full screen window 904 covers both displays 110,114 and the first touch sensitive display 110 is on the primary screen 104. Thus, the configurable area 112 of the primary screen 104 is provided in the configuration shown in Fig. 9A.

[0109] Afrer focus is established, an orientation change 908 may be received to change the device from a open portrait state 320 to open landscape state 348, in step 1016. For example, a user may turn the device 100 in the direction of the arrow 908 to change the orientation of the device 100. As shown in Fig. 9B, the device 100 is now in landscape orientation. In embodiments, the Window Management Module 532 maintains focus on window 1 904 based on information provided from the Task Management Module 540, in step 1020. However, the configurable area configuration is changed to provide configurable area 116 instead of configurable area 112 (as represented by arrow 912), in step 1024. The configurable area 116 can be moved in the configuration shown in Fig. 9B to give the user more area to hold the device 100 around the first touch sensitive display 110 at the bottom of the device 100. Thus, the

Window Management Module 532 determines that the configurable area configuration should change based on information provided from the Task Management Module 540, even though the focus did not change from window 1 904.

[0110] An embodiment of another method 1200 for changing focus is shown in Fig. 12. While a general order for the steps of the method 1200 is shown in Fig. 12. Generally, the method 1200 starts with a start operation 1204 and ends with an end operation 1228. The method 1200 can include more or fewer steps or can arrange the order of the steps differently than those shown in Fig. 12. The method 1200 can be executed as a set of computer-executable instructions executed by a computer system and encoded or stored on a computer readable medium. Hereinafrer, the method 1200 shall be explained with reference to the systems, components, modules, software, data structures, user interfaces, etc. described in conjunction with Figs. 1-11B.

[0111] In embodiments, the device 120 can establish focus in the user interface, in step 1208. However, it is possible that focus is not established on a window but appears to be on the desktop. For example, the device 120 may not have set the focus on window 1 1108 but may make it appear that the desktop 1104 has focus by having the configurable area 112 provided rather than configurable area 116. The active window 1 1108 may occupy only the first touch sensitive display 114 and not the first touch sensitive display 110. In embodiments, the Window Management Module 532 establishes focus based on information provided from the Task Management Module 540. In response to the focus establishment, the device 120 may also determine which configurable area 112,116 is provide or configured, in step 1212. In the embodiment shown in Fig. 11A, configurable area 112 is provided as window 1 1108 does not have focus and the first touch sensitive display 112 is on the primary screen 124. Thus, the configurable area 112 of the primary screen 124 is provided in the configuration shown in Fig. 11A.

[0112] Afrer focus is established, an orientation change 1112 may be received to change the device from a open portrait state 320 to open landscape state 348, in step 1216. For example, a user may turn the device 120 in the direction of the arrow 1112 to change the orientation of the device 120. As shown in Fig. 11B, the device 120 is now in landscape orientation. In embodiments, the Window Management Module 532 or some other component may change the display of the active window 1108 by expanding the window 1108 into a full screen window by

displaying an action called a "gravity drop." A gravity drop appears to make the bottom part of the window 1108 drop to the bottom part of touch sensitive display 110. Thus, the active window 1108 becomes a full screen window. After the window 1108 occupies the entire display area, the Window Management Module 532 may change the focus to window 1 1108 based on information provided from the Task Management Module 540, in step 1220.

[0113] However, the configurable area configuration is also changed to provide configurable area 116 instead of configurable area 112 (as represented by arrow 1116), in step 1224. The configurable area 116 can be moved in the configuration shown in Fig. 11B to give the user more area to hold the device 120 around the first touch sensitive display 110 at the bottom of the device 120. Thus, the Window Management Module 532 determines that the configurable area configuration should change based on information provided from the Task Management Module 540, even though the focus did not change from window 1 1108.

[0114] The exemplary systems and methods of this disclosure have been described in relation to a window stack associated with a multi-screen device. However, to avoid unnecessarily obscuring the present disclosure, the preceding description omits a number of known structures and devices. This omission is not to be construed as a limitation of the scopes of the claims. Specific details are set forth to provide an understanding of the present disclosure. It should however be appreciated that the present disclosure may be practiced in a variety of ways beyond the specific detail set forth herein.

[0115] Furthermore, while the exemplary aspects, embodiments, and/or configurations illustrated herein show the various components of the system collocated, certain components of the system can be located remotely, at distant portions of a distributed network, such as a LAN and/or the Internet, or within a dedicated system. Thus, it should be appreciated, that the components of the system can be combined in to one or more devices, such as the software modules of Figs. 5A and 5B, or collocated on a particular node of a distributed network, such as an analog and/or digital telecommunications network, a packet-switch network, or a circuit-switched network. It will be appreciated from the preceding description, and for reasons of computational efficiency, that the components of the system can be arranged at any location within a distributed network of components without affecting the operation of the system. For example, the various components can be located in a switch such as a PBX and media server, gateway, in one or more communications devices, at one or more users' premises, or some

combination thereof. Similarly, one or more functional portions of the system could be distributed between a telecommunications device(s) and an associated computing device.

[0116] Furthermore, it should be appreciated that the various links connecting the elements can be wired or wireless links, or any combination thereof, or any other known or later developed element(s) that is capable of supplying and/or communicating data to and from the connected elements. These wired or wireless links can also be secure links and may be capable of communicating encrypted information. Transmission media used as links, for example, can be any suitable carrier for electrical signals, including coaxial cables, copper wire and fiber optics, and may take the form of acoustic or light waves, such as those generated during radio-wave and infra-red data communications.

[0117] Also, while the flowcharts have been discussed and illustrated in relation to a particular sequence of events, it should be appreciated that changes, additions, and omissions to this sequence can occur without materially affecting the operation of the disclosed embodiments, configuration, and aspects.

[0118] A number of variations and modifications of the disclosure can be used. It would be possible to provide for some features of the disclosure without providing others. For example in one alternative embodiment, the window stack 700 resembles a carousel or rolodex rather than a deck of cards. Thus, windows may circulate from one touch sensitive display 110 to another touch sensitive display 114. For example, window may be push to the right and end up at the end of stack behind another window. If the stack continues to move to the right, eventually the window would appear on the second touch sensitive display even though the window was opened in the first touch sensitive display 110. These movements and changes in the stacks may be managed using the methods and logical data structures discussed above. In another alternative embodiment, there may be other arrangement for the window stack.

[0119] In yet another embodiment, the systems and methods of this disclosure can be implemented in conjunction with a special purpose computer, a programmed microprocessor or microcontroller and peripheral integrated circuit element(s), an ASIC or other integrated circuit, a digital signal processor, a hard-wired electronic or logic circuit such as discrete element circuit, a programmable logic device or gate array such as PLD, PLA, FPGA, PAL, special purpose computer, any comparable means, or the like. In general, any device(s) or means capable of implementing the methodology illustrated herein can be used to implement the various aspects of

this disclosure. Exemplary hardware that can be used for the disclosed embodiments, configurations and aspects includes computers, handheld devices, telephones (e.g., cellular, Internet enabled, digital, analog, hybrids, and others), and other hardware known in the art. Some of these devices include processors (e.g., a single or multiple microprocessors), memory, nonvolatile storage, input devices, and output devices. Furthermore, alternative software implementations including, but not limited to, distributed processing or component/object distributed processing, parallel processing, or virtual machine processing can also be constructed to implement the methods described herein.

[0120] In yet another embodiment, the disclosed methods may be readily implemented in conjunction with software using object or object-oriented software development environments that provide portable source code that can be used on a variety of computer or workstation platforms. Alternatively, the disclosed system may be implemented partially or fully in hardware using standard logic circuits or VLSI design. Whether software or hardware is used to implement the systems in accordance with this disclosure is dependent on the speed and/or efficiency requirements of the system, the particular function, and the particular software or hardware systems or microprocessor or microcomputer systems being utilized.

[0121] In yet another embodiment, the disclosed methods may be partially implemented in software that can be stored on a storage medium, executed on programmed general-purpose computer with the cooperation of a controller and memory, a special purpose computer, a microprocessor, or the like. In these instances, the systems and methods of this disclosure can be implemented as program embedded on personal computer such as an applet, JAVA® or CGI script, as a resource residing on a server or computer workstation, as a routine embedded in a dedicated measurement system, system component, or the like. The system can also be implemented by physically incorporating the system and/or method into a software and/or hardware system.

[0122] Although the present disclosure describes components and functions implemented in the aspects, embodiments, and/or configurations with reference to particular standards and protocols, the aspects, embodiments, and/or configurations are not limited to such standards and protocols. Other similar standards and protocols not mentioned herein are in existence and are considered to be included in the present disclosure. Moreover, the standards and protocols mentioned herein and other similar standards and protocols not mentioned herein are periodically

superseded by faster or more effective equivalents having essentially the same functions. Such replacement standards and protocols having the same functions are considered equivalents included in the present disclosure.

[0123] The present disclosure, in various aspects, embodiments, and/or configurations, includes components, methods, processes, systems and/or apparatus substantially as depicted and described herein, including various aspects, embodiments, configurations embodiments, subcombinations, and/or subsets thereof. Those of skill in the art will understand how to make and use the disclosed aspects, embodiments, and/or configurations afrer understanding the present disclosure. The present disclosure, in various aspects, embodiments, and/or configurations, includes providing devices and processes in the absence of items not depicted and/or described herein or in various aspects, embodiments, and/or configurations hereof, including in the absence of such items as may have been used in previous devices or processes, e.g., for improving performance, achieving ease and\or reducing cost of implementation.

[0124] The foregoing discussion has been presented for purposes of illustration and description. The foregoing is not intended to limit the disclosure to the form or forms disclosed herein. In the foregoing Detailed Description for example, various features of the disclosure are grouped together in one or more aspects, embodiments, and/or configurations for the purpose of streamlining the disclosure. The features of the aspects, embodiments, and/or configurations of the disclosure may be combined in alternate aspects, embodiments, and/or configurations other than those discussed above. This method of disclosure is not to be interpreted as reflecting an intention that the claims require more features than are expressly recited in each claim. Rather, as the following claims reflect, inventive aspects lie in less than all features of a single foregoing disclosed aspect, embodiment, and/or configuration. Thus, the following claims are hereby incorporated into this Detailed Description, with each claim standing on its own as a separate preferred embodiment of the disclosure.

[0125] Moreover, though the description has included description of one or more aspects, embodiments, and/or configurations and certain variations and modifications, other variations, combinations, and modifications are within the scope of the disclosure, e.g., as may be within the skill and knowledge of those in the art, afrer understanding the present disclosure. It is intended to obtain rights which include alternative aspects, embodiments, and/or configurations to the extent permitted, including alternate, interchangeable and/or equivalent structures, functions,

ranges or steps to those claimed, whether or not such alternate, interchangeable and/or equivalent structures, functions, ranges or steps are disclosed herein, and without intending to publicly dedicate any patentable subject matter.

What is claimed is:

1. A computer readable medium, having stored thereon, computer-executable instructions executable by a processor, the computer-executable instructions causing the processor to execute a method for changing focus in a multi-screen device, the computer-executable instructions comprising:

instructions to establish focus on a window;

instructions to receive a user input to reveal a desktop;

instructions to remove focus from the window;

instructions to hide the desktop;

instructions to again display the window; and

instructions to place focus back on the window.

2. The computer readable medium as defined in claim 1, wherein the user input is a gesture.

3. The computer readable medium as defined in claim 2, wherein the gesture is a spread gesture.

4. The computer readable medium as defined in claim 2, wherein, before the desktop is revealed, a configurable area is provided, the configurable area is associated with a display displaying the window..

5. The computer readable medium as defined in claim 4, wherein the desktop is displayed across substantially all of at least the two or more touch sensitive displays.

6. The computer readable medium as defined in claim 5, wherein when the desktop is revealed, the window appears to move from a display.

7. The computer readable medium as defined in claim 6, wherein hiding the desktop is in response to a second gesture.

8. The computer readable medium as defined in claim 6, wherein the window is not on a primary screen.

9. The computer readable medium as defined in claim 8, wherein a second window is display with the window when the desktop is hidden.

10. A device comprising:

at least two screens, each screen comprising:

a touch sensitive display;

a configurable area;

wherein a first display displays a first window and a second display displays a second window;

a memory;

a processor in communication with the memory and each of the at least two screens, the processor operable to:

establish focus on the second window;

receive an input from the second configurable area to reveal a desktop;

remove focus from the second window;

hide the first and second window;

reveal the desktop;

receive a gesture in either the first or second touch sensitive display to hide the desktop;

display the first and second window; and

place focus back on the second window.

11. The device as defined in claim 10, wherein the second configurable area remains active.

12. The device as defined in claim 11, further comprising the first display is on a primary screen.

13. The device as defined in claim 12, wherein the gesture is a spread gesture.

14. The device as defined in claim 13, wherein, when the first and second window are hidden, the first and second window appear to slide of the first and second displays.

15. The device as defined in claim 14, wherein the desktop appears to be exposed from behind the first and second window when revealed.

16. The device as defined in claim 15, wherein it appears to the user that the second window maintained focus during the desktop reveal.

17. A method for presenting a display for a multi-screen device, the method comprising:

establishing focus on a first window;

changing focus to a second window;

receiving an input ftom a second configurable area to reveal a desktop;

removing focus ftom the second window;

moving the first and second window ftom a first and second display;

revealing the desktop;

receiving a gesture to hide the desktop;

redisplaying the first and second window; and

replacing focus back on the second window.

18. The method defined in claim 17, wherein the desktop is behind the first and second window.

19. The method defined in claim 18, wherein the first window is displayed on a primary screen.

20. The method defined in claim 19, wherein the second configurable area is maintained.

ABSTRACT

Systems and methods are provides for adjusting focus during a desktop reveal. A window has focus before the desktop is revealed. After the window is returned and the desktop hidden, the focus is again placed on the window. Further, a configurable area associated with the screen that displays the window is maintained during the desktop reveal and the return of the window.

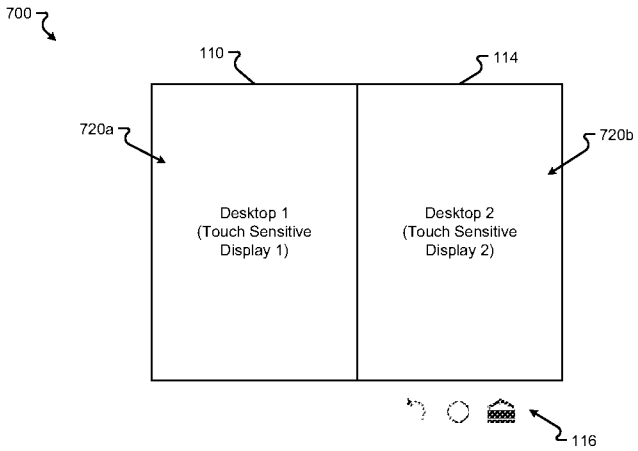


FIG. 7C

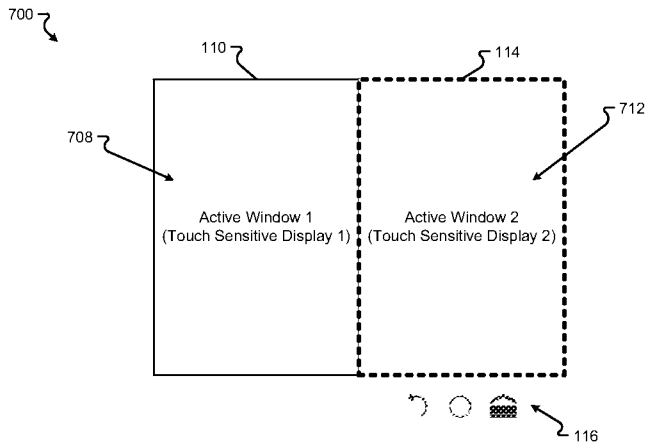


FIG. 7A

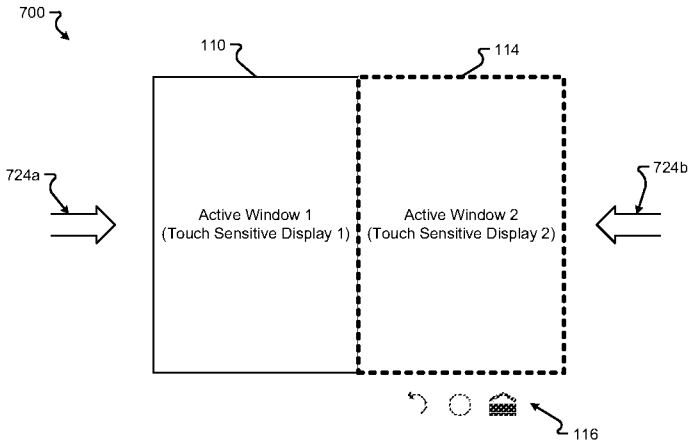


FIG. 7D

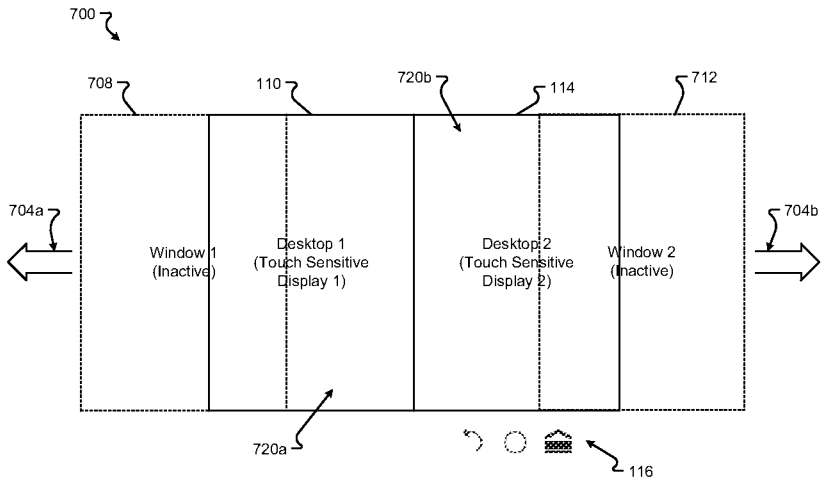


FIG. 7B

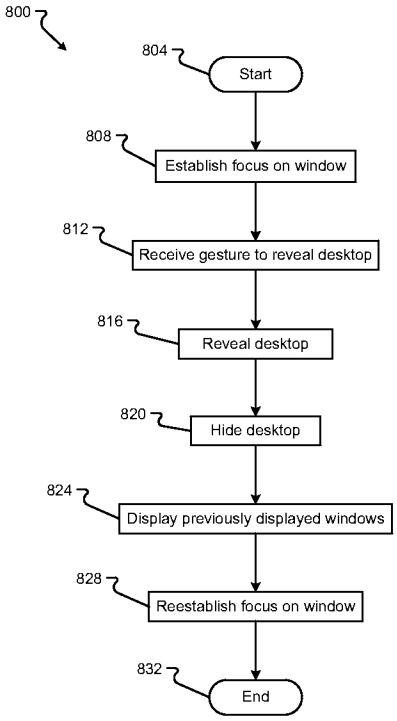


FIG. 8

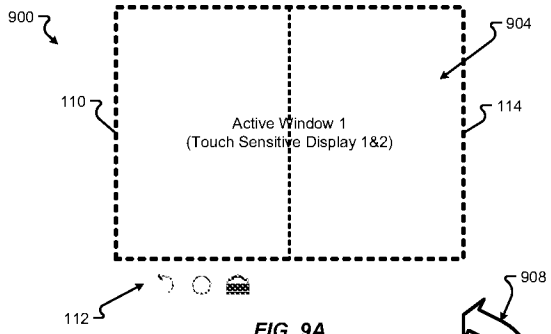


FIG. 9A

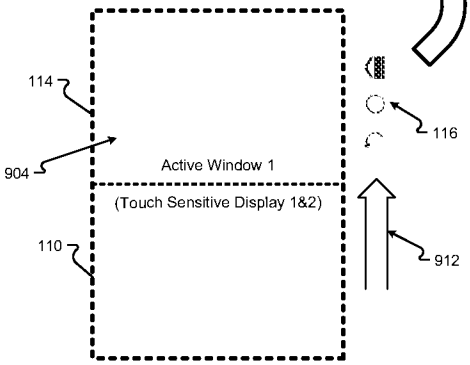


FIG. 9B

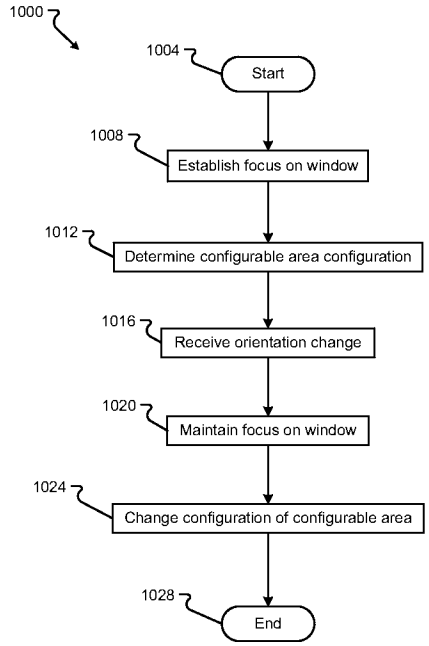


FIG. 10

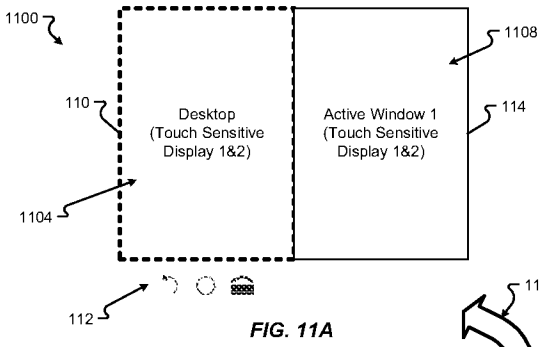


FIG. 11A

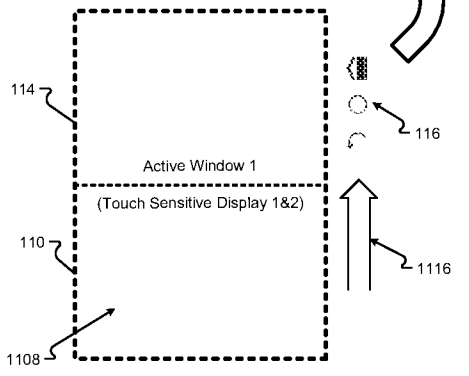


FIG. 11B

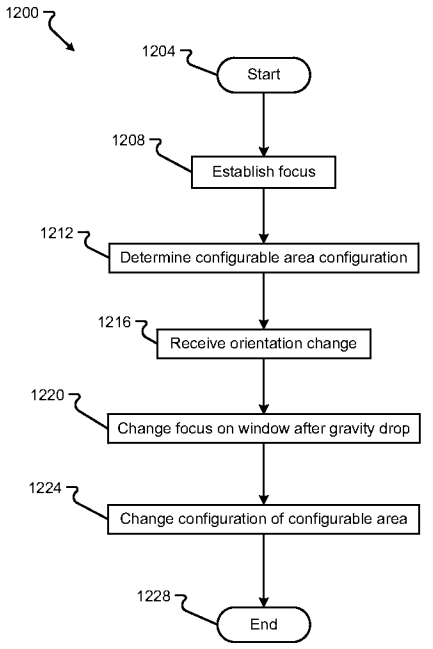

1200
1204
Start
1208
Establish focus
1212
Determine configurable area configuration
1216
Receive orientation change
1220
Change focus on window after gravity drop
1224
Change configuration of configurable area
1228
End

FIG. 12

[0100] The device 100 manages desktops and/or windows with at least one window stack 700, 728, as shown in Figs. 7A and 7B. A window stack 700, 728 is a logical arrangement of active and/or inactive windows for a multi-screen device. For example, the window stack 700, 728 may be logically similar to a deck of cards, where one or more windows or desktops are arranged in order, as shown in Figs. 7A and 7B. An active window is a window that is currently being displayed on at least one of the touch sensitive displays 110, 114. For example, windows 104 and 108 are active windows and are displayed on touch sensitive displays 110 and 114. An inactive window is a window that was opened and displayed but is now "behind" an active window and not being displayed. In embodiments, an inactive window may be for an application that is suspended, and thus, the window is not displaying active content. For example, windows 712, 716, 720, and 724 are inactive windows.

[0101] A window stack 700, 728 may have various arrangements or organizational structures. In the embodiment shown in Fig. 7A, the device 100 includes a first stack 760 associated with a first touch sensitive display 110 and a second stack associated with a second touch sensitive display 114. Thus, each touch sensitive display 110, 114 can have an associated window stack 760, 764. These two window stacks 760, 764 may have different numbers of windows arranged in the respective stacks 760, 764. Further, the two window stacks 760, 764 can also be identified differently and managed separately. Thus, the first window stack 760 can be arranged in order from a first window 704 to a next window 720 to a last window 724 and finally to a desktop 722, which, in embodiments, is at the "bottom" of the window stack 760. In embodiments, the desktop 722 is not always at the "bottom" as application windows can be arranged in the window stack below the desktop 722, and the desktop 722 can be brought to the "top" of a stack over other windows during a desktop reveal. Likewise, the second stack 764 can be arranged from a first window 708 to a next window 712 to a last window 716, and finally to a desktop 718, which, in embodiments, is a single desktop area, with desktop 722, under all the windows in both window stack 760 and window stack 764. A logical data structure for managing the two window stacks 760, 764 may be as described in conjunction with Fig. 8.

[0102] Another arrangement for a window stack 728 is shown in Fig. 7B. In this embodiment, there is a single window stack 728 for both touch sensitive displays 110, 114. Thus, the window stack 728 is arranged from a desktop 758 to a first window 744 to a last window 756. A window can be arranged in a position among all windows without an association to a specific touch

sensitive display 110, 114. In this embodiment, a window is in the order of windows. Further, at least one window is identified as being active. For example, a single window may be rendered in two portions 732 and 736 that are displayed on the first touch sensitive screen 110 and the second touch sensitive screen 114. The single window may only occupy a single position in the window stack 728 although it is displayed on both displays 110, 114.

[0103] Yet another arrangement of a window stack 760 is shown in Figs. 7C through 7E. The window stack 760 is shown in three "elevation" views. In Fig. 7C, the top of the window stack 760 is shown. Two sides of the window stack 760 are shown in Figs. 7D and 7E. In this embodiment, the window stack 760 resembles a stack of bricks. The windows are stacked on each other. Looking from the top of the window stack 760 in Fig, 7C, only the top most windows in the window stack 760 are seen in different portions of the composite display 764. The composite display 764 represents a logical model for the entire display area of the device 100, which can include touch sensitive display 110 and touch sensitive display 114. A desktop 786 or a window can occupy part or all of the composite display 764.

[0104] In the embodiment shown, the desktop 786 is the lowest display or "brick" in the window stack 760. Thereupon, window 1 782, window 2 782, window 3 768, and window 4 770 are layered. Window 1 782, window 3 768, window 2 782, and window 4 770 only occupy a portion of the composite display 764. Thus, another part of the stack 760 includes window 8 774 and windows 5 through 7 shown in section 790. Only the top window in any portion of the composite display 764 is actually rendered and displayed. Thus, as shown in the top view in Fig. 7C, window 4 770, window 8 774, and window 3 768 are displayed as being at the top of the display in different portions of the window stack 760. A window can be dimensioned to occupy only a portion of the composite display 760 to "reveal" windows lower in the window stack 760. For example, window 3 768 is lower in the stack than both window 4 770 and window 8 774 but is still displayed. A logical data structure to manage the window stack can be as described in conjunction with Fig. 8.

[0105] When a new window is opened, the newly activated window is generally positioned at the top of the stack. However, where and how the window is positioned within the stack can be a function of the orientation of the device 100, the context of what programs, functions, software, etc. are being executed on the device 100, how the stack is positioned when the new window is opened, etc. To insert the window in the stack, the position in the stack for the window is

determined and the touch sensitive display 110, 114 to which the window is associated may also be determined. With this information, a logical data structure for the window can be created and stored. When user interface or other events or tasks change the arrangement of windows, the window stack(s) can be changed to reflect the change in arrangement. It should be noted that these same concepts described above can be used to manage the one or more desktops for the device 100.

[0106] A logical data structure 800 for managing the arrangement of windows or desktops in a window stack is shown in Fig. 8. The logical data structure 800 can be any data structure used to store data whether an object, record, file, etc. The logical data structure 800 can be stored in any type of database or data storage system, regardless of protocol or standard. In embodiments, the logical data structure 800 includes one or more portions, fields, attributes, etc. that store data in a logical arrangement that allows for easy storage and retrieval of the information. Hereinafrer, these one or more portions, fields, attributes, etc. shall be described simply as fields. The fields can store data for a window identifier 804, dimensions 808, a stack position identifier 812, a display identifier 816, and/or an active indicator 820. Each window in a window stack can have an associated logical data structure 800. While only a single logical data structure 800 is shown in Fig. 8, there may be more or fewer logical data structures 800 used with a window stack (based on the number of windows or desktops in the stack), as represented by ellipses 824. Further, there may be more or fewer fields than those shown in Fig. 8, as represented by ellipses 828.

[0107] A window identifier 804 can include any identifier (ID) that uniquely identifies the associated window in relation to other windows in the window stack. The window identifier 804 can be a globally unique identifier (GUID), a numeric ID, an alphanumeric ID, or other type of identifier. In embodiments, the window identifier 804 can be one, two, or any number of digits based on the number of windows that can be opened. In alternative embodiments, the size of the window identifier 804 may change based on the number of windows opened. While the window is open, the window identifier 804 may be static and remain unchanged.

[0108] Dimensions 808 can include dimensions for a window in the composite display 760. For example, the dimensions 808 can include coordinates for two or more corners of the window or may include one coordinate and dimensions for the width and height of the window. These dimensions 808 can delineate what portion of the composite display 760 the window may

occupy, which may the entire composite display 760 or only part of composite display 760. For example, window 4 770 may have dimensions 880 that indicate that the window 770 will occupy only part of the display area for composite display 760, as shown in Figs. 7c through 7E. As windows are moved or inserted in the window stack, the dimensions 808 may change.

[0109] A stack position identifier 812 can be any identifier that can identify the position in the stack for the window or may be inferred from the window's control record within a data structure, such as a list or a stack. The stack position identifier 812 can be a GUID, a numeric ID, an alphanumeric ID, or other type of identifier. Each window or desktop can include a stack position identifier 812. For example, as shown in Fig. 7A, window 1 704 in stack 1 760 can have a stack position identifier 812 of 1 identifying that window 704 is the first window in the stack 760 and the active window. Similarly, window 6 724 can have a stack position identifier 812 of 3 representing that window 724 is the third window in the stack 760. Window 2 708 can also have a stack position identifier 812 of 1 representing that window 708 is the first window in the second stack 764. As shown in Fig. 7B, window 1 744 can have a stack position identifier 812 of 1, window 3, rendered in portions 732 and 736, can have a stack position identifier 812 of 3, and window 6 756 can have a stack position identifier 812 of 6. Thus, depending on the type of stack, the stack position identifier 812 can represent a window's location in the stack.

[0110] A display identifier 816 can identify that the window or desktop is associated with a particular display, such as the first display 110 or the second display 114, or the composite display 760 composed of both displays. While this display identifier 816 may not be needed for a multi-stack system, as shown in Fig. 7A, the display identifier 816 can indicate whether a window in the serial stack of Fig. 7B is displayed on a particular display. Thus, window 3 may have two portions 732 and 736 in Fig. 7B. The first portion 732 may have a display identifier 816 for the first display while the second portion 736 may have a display identifier 816 for the second display 114. However, in alternative embodiments, the window may have two display identifier 816 that represent that the window is displayed on both of the displays 110, 114, or a display identifier 816 identifying the composite display. In another alternate embodiment, the window may have a single display identifier 816 to represent that the window is displayed on both of the displays 110, 114.

[0111] Similar to the display identifier 816, an active indicator 820 may not be needed with the dual stack system of Fig. 7A, as the window in stack position 1 is active and displayed. In the

system of Fig. 7B, the active indicator 820 can indicate which window(s) in the stack is being displayed. Thus, window 3 may have two portions 732 and 736 in Fig. 7. The first portion 732 may have an active indicator 820 while the second portion 736 may also have an active indicator 820. However, in alternative embodiments, window 3 may have a single active indicator 820. The active indicator 820 can be a simple flag or bit that represents that the window is active or displayed.

[0112] An embodiment of a method 900 for creating a window stack is shown in Fig. 9. While a general order for the steps of the method 900 is shown in Fig. 9. Generally, the method 900 starts with a start operation 904 and ends with an end operation 928. The method 900 can include more or fewer steps or can arrange the order of the steps differently than those shown in Fig. 9. The method 900 can be executed as a set of computer-executable instructions executed by a computer system and encoded or stored on a computer readable medium. Hereinafter, the method 900 shall be explained with reference to the systems, components, modules, software, data structures, user interfaces, etc. described in conjunction with Figs. 1-8.

[0113] A multi-screen device 100 can receive activation of a window, in step 908. In embodiments, the multi-screen device 100 can receive activation of a window by receiving an input from the touch sensitive display 110 or 114, the configurable area 112 or 116, a gesture capture region 120 or 124, or some other hardware sensor operable to receive user interface inputs. The processor may execute the Task Management Module 540 may receive the input. The Task Management Module 540 can interpret the input as requesting an application task to be executed that will open a window in the window stack.

[0114] In embodiments, the Task Management Module 540 places the user interface interaction in the task stack 552 to be acted upon by the Display Configuration Module 568 of the Multi-Display Management Module 524. Further, the Task Management Module 540 waits for information from the Multi-Display Management Module 524 to send instructions to the Window Management Module 532 to create the window in the window stack.

[0115] The Multi-Display Management Module 524, upon receiving instruction from the Task Management Module 540, determines to which touch portion of the composite display 760, the newly activated window should be associated, in step 912. For example, window 4 770 is associated with the a portion of the composite display 764 In embodiments, the device state module 574 of the Multi-Display Management Module 524 may determine how the device is

oriented or in what state the device is in, e.g., open, closed, portrait, etc. Further, the preferences module 572 and/or requirements module 580 may determine how the window is to be displayed. The gesture module 576 may determine the user's intentions about how the window is to be opened based on the type of gesture and the location of where the gesture is made.

[0116] The Display Configuration Module 568 may use the input from these modules and evaluate the current window stack 760 to determine the best place and the best dimensions, based on a visibility algorithm, to open the window. Thus, the Display Configuration Module 568 determines the best place to put the window at the top of the window stack 760, in step 916. The visibility algorithm, in embodiments, determines for all portions of the composite display, which windows are at the top of the stack. For example, the visibility algorithm determines that window 3 768, window 4 770, and window 8 774 are at the top of the stack 760 as viewed in Figs. 7C through 7E. Upon determining where to open the window, the Display Configuration Module 568 can assign a display identifier 816 and possibly dimensions 808 to the window. The display identifier 816 and dimensions 808 can then be sent back to the Task Management Module 540. The Task Management Module 540 may then assign the window a stack position identifier 812 indicating the windows position at the top of the window stack.

[0117] In embodiments, the Task Management Module 540 sends the window stack information and instructions to render the window to the Window Management Module 532. The Window Management Module 532 and the Task Management Module 540 can create the logical data structure 800, in step 924. Both the Task Management Module 540 and the Window Management Module 532 may create and manage copies of the window stack. These copies of the window stack can be synchronized or kept similar through communications between the Window Management Module 532 and the Task Management Module 540. Thus, the Window Management Module 532 and the Task Management Module 540, based on the information determined by the Multi-Display Management Module 524, can assign dimensions 808, a stack position identifier 812 (e.g., window 1 782, window 4 770, etc.), a display identifier 816 (e.g., touch sensitive display 1 110, touch sensitive display 2 114, composite display identifier, etc,), and an active indicator 820, which is generally always set when the window is at the "top" of the stack. The logical data structure 800 may then be stored by both the Window Management Module 532 and the Task Management Module 540. Further, the Window Management Module

532 and the Task Management Module 540may thereinafrer manage the window stack and the logical data structure(s) 800.

[0118] The exemplary systems and methods of this disclosure have been described in relation to a window stack associated with a multi-screen device. However, to avoid unnecessarily obscuring the present disclosure, the preceding description omits a number of known structures and devices. This omission is not to be construed as a limitation of the scopes of the claims. Specific details are set forth to provide an understanding of the present disclosure. It should however be appreciated that the present disclosure may be practiced in a variety of ways beyond the specific detail set forth herein.

[0119] Furthermore, while the exemplary aspects, embodiments, and/or configurations illustrated herein show the various components of the system collocated, certain components of the system can be located remotely, at distant portions of a distributed network, such as a LAN and/or the Internet, or within a dedicated system. Thus, it should be appreciated, that the components of the system can be combined in to one or more devices, such as the software modules of Figs. 5A and 5B, or collocated on a particular node of a distributed network, such as an analog and/or digital telecommunications network, a packet-switch network, or a circuit-switched network. It will be appreciated from the preceding description, and for reasons of computational efficiency, that the components of the system can be arranged at any location within a distributed network of components without affecting the operation of the system. For example, the various components can be located in a switch such as a PBX and media server, gateway, in one or more communications devices, at one or more users' premises, or some combination thereof. Similarly, one or more functional portions of the system could be distributed between a telecommunications device(s) and an associated computing device.

[0120] Furthermore, it should be appreciated that the various links connecting the elements can be wired or wireless links, or any combination thereof, or any other known or later developed element(s) that is capable of supplying and/or communicating data to and from the connected elements. These wired or wireless links can also be secure links and may be capable of communicating encrypted information. Transmission media used as links, for example, can be any suitable carrier for electrical signals, including coaxial cables, copper wire and fiber optics, and may take the form of acoustic or light waves, such as those generated during radio-wave and infra-red data communications.

[0121] Also, while the flowcharts have been discussed and illustrated in relation to a particular sequence of events, it should be appreciated that changes, additions, and omissions to this sequence can occur without materially affecting the operation of the disclosed embodiments, configuration, and aspects.

[0122] A number of variations and modifications of the disclosure can be used. It would be possible to provide for some features of the disclosure without providing others. For example in one alternative embodiment, the window stack 700 resembles a carousel or rolodex rather than a deck of cards. Thus, windows may circulate from one touch sensitive display 110 to another touch sensitive display 114. For example, window 6 724 may be pushed to the right and end up at the end of stack 764 behind window 716. If the stack continues to move to the right, eventually window 6 724 would appear on the second touch sensitive display 114 even though the window 724 was opened in the first touch sensitive display 110. In another alternative embodiment, there may be other arrangement for the window stack.

[0123] In yet another embodiment, the systems and methods of this disclosure can be implemented in conjunction with a special purpose computer, a programmed microprocessor or microcontroller and peripheral integrated circuit element(s), an ASIC or other integrated circuit, a digital signal processor, a hard-wired electronic or logic circuit such as discrete element circuit, a programmable logic device or gate array such as PLD, PLA, FPGA, PAL, special purpose computer, any comparable means, or the like. In general, any device(s) or means capable of implementing the methodology illustrated herein can be used to implement the various aspects of this disclosure. Exemplary hardware that can be used for the disclosed embodiments, configurations and aspects includes computers, handheld devices, telephones (e.g., cellular, Internet enabled, digital, analog, hybrids, and others), and other hardware known in the art. Some of these devices include processors (e.g., a single or multiple microprocessors), memory, nonvolatile storage, input devices, and output devices. Furthermore, alternative software implementations including, but not limited to, distributed processing or component/object distributed processing, parallel processing, or virtual machine processing can also be constructed to implement the methods described herein.

[0124] In yet another embodiment, the disclosed methods may be readily implemented in conjunction with software using object or object-oriented software development environments that provide portable source code that can be used on a variety of computer or workstation

platforms. Alternatively, the disclosed system may be implemented partially or fully in hardware using standard logic circuits or VLSI design. Whether software or hardware is used to implement the systems in accordance with this disclosure is dependent on the speed and/or efficiency requirements of the system, the particular function, and the particular software or hardware systems or microprocessor or microcomputer systems being utilized.

[0125] In yet another embodiment, the disclosed methods may be partially implemented in software that can be stored on a storage medium, executed on programmed general-purpose computer with the cooperation of a controller and memory, a special purpose computer, a microprocessor, or the like. In these instances, the systems and methods of this disclosure can be implemented as program embedded on personal computer such as an applet, JAVA® or CGI script, as a resource residing on a server or computer workstation, as a routine embedded in a dedicated measurement system, system component, or the like. The system can also be implemented by physically incorporating the system and/or method into a software and/or hardware system.

[0126] Although the present disclosure describes components and functions implemented in the aspects, embodiments, and/or configurations with reference to particular standards and protocols, the aspects, embodiments, and/or configurations are not limited to such standards and protocols. Other similar standards and protocols not mentioned herein are in existence and are considered to be included in the present disclosure. Moreover, the standards and protocols mentioned herein and other similar standards and protocols not mentioned herein are periodically superseded by faster or more effective equivalents having essentially the same functions. Such replacement standards and protocols having the same functions are considered equivalents included in the present disclosure.

[0127] The present disclosure, in various aspects, embodiments, and/or configurations, includes components, methods, processes, systems and/or apparatus substantially as depicted and described herein, including various aspects, embodiments, configurations embodiments, subcombinations, and/or subsets thereof. Those of skill in the art will understand how to make and use the disclosed aspects, embodiments, and/or configurations after understanding the present disclosure. The present disclosure, in various aspects, embodiments, and/or configurations, includes providing devices and processes in the absence of items not depicted and/or described herein or in various aspects, embodiments, and/or configurations hereof,

including in the absence of such items as may have been used in previous devices or processes, e.g., for improving performance, achieving ease and\or reducing cost of implementation.

[0128] The foregoing discussion has been presented for purposes of illustration and description. The foregoing is not intended to limit the disclosure to the form or forms disclosed herein. In the foregoing Detailed Description for example, various features of the disclosure are grouped together in one or more aspects, embodiments, and/or configurations for the purpose of streamlining the disclosure. The features of the aspects, embodiments, and/or configurations of the disclosure may be combined in alternate aspects, embodiments, and/or configurations other than those discussed above. This method of disclosure is not to be interpreted as reflecting an intention that the claims require more features than are expressly recited in each claim. Rather, as the following claims reflect, inventive aspects lie in less than all features of a single foregoing disclosed aspect, embodiment, and/or configuration. Thus, the following claims are hereby incorporated into this Detailed Description, with each claim standing on its own as a separate preferred embodiment of the disclosure.

[0129] Moreover, though the description has included description of one or more aspects, embodiments, and/or configurations and certain variations and modifications, other variations, combinations, and modifications are within the scope of the disclosure, e.g., as may be within the skill and knowledge of those in the art, after understanding the present disclosure. It is intended to obtain rights which include alternative aspects, embodiments, and/or configurations to the extent permitted, including alternate, interchangeable and/or equivalent structures, functions, ranges or steps to those claimed, whether or not such alternate, interchangeable and/or equivalent structures, functions, ranges or steps are disclosed herein, and without intending to publicly dedicate any patentable subject matter.

What is claimed is:

1. A computer readable medium, having stored thereon, computer-executable instructions executable by a processor, the computer-executable instructions causing the processor to execute a method for creating a window stack for a multi-screen device, the computer-executable instructions comprising:

instructions to receive activation of a window in the multi-screen device;

instructions to determine a display associated with the active window;

instructions to determine a position in a window stack for the active window; and

after determining the position and display of the active window, instructions to generate a logic data structure for the active window to describe the position of the active window in the window stack.

2. The computer readable medium as defined in claim 1, wherein the window stack is a logical arrangement of active and inactive windows for a multi-screen device.

3. The computer readable medium as defined in claim 2, wherein an active window is displayed on at least one of two displays of the multi-screen device.

4. The computer readable medium as defined in claim 1, wherein the position in the window stack can be associated with at least one of two displays of the multi-screen device.

5. The computer readable medium as defined in claim 1, wherein the position in the window stack can be a position among all windows without association to one of the displays of the multi-screen device.

6. The computer readable medium as defined in claim 1, wherein the logical data structure comprises one or more of:

a window identifier adapted to identify the active window in relation to other windows in the window stack;

a window stack position identifier adapted to identify the position in the window stack for the active window; and

a display identifier adapted to identify which one of two displays of the multi-screen device the active window is associated.

7. The computer readable medium as defined in claim 6, wherein a first display of the multi-screen device is associated with a first stack and a second display of the multi-screen device is associated with a second stack.

8. The computer readable medium as defined in claim 7, wherein the logic data structure further comprises dimensions that delineate a portion of a display that the active window will occupy.

9. The computer readable medium as defined in claim 8, wherein a single window is displayed across both the first and second display.

10. The computer readable medium as defined in claim 9, wherein the single window has an identifier of a composite display, which includes both the first and second display.

11. A multi-screen device comprising:

two displays;

a memory;

a processor in communication with the memory and each of the two displays, the processor operable to:

receive activation of a window in the multi-screen device;

determine at least one of the displays of the multi-screen device associated with the active window;

determine a position in a window stack for the active window, wherein the active window is displayed on at least one of displays of the multi-screen device; and

after determining the window stack position and associated display of the active window, generate a logic data structure for the active window to describe the position of the active window in the window stack, wherein the window stack is an arrangement of a desktop and at least one other window displayed in at least one of the displays.

12. The computer readable medium as defined in claim 11, wherein the stack includes at least one active window, wherein the active window is positioned on top of the window stack and is displayed on at least one of two displays of the multi-screen device.

13. The computer readable medium as defined in claim 11, wherein the position in the window stack can be associated with a composite display, which includes all of the displays of the multi-screen device.

14. The computer readable medium as defined in claim 11, wherein the logic data structure comprises one or more of:

a window identifier adapted to identify the active window in relation to other windows in the window stack;

a window stack position identifier adapted to identify the position in the window stack for the active window; and

a display identifier adapted to identify which one of the displays of the multi-screen device the active window is associated.

15. The computer readable medium as defined in claim 14, wherein a first display of the multi-screen device is associated with a first stack and a second display of the multi-screen device is associated with a second stack.

16. A method for creating a window stack in a multi-screen device, the method comprising:

a processor executing a framework, a desktop, and an Application Manager;

the desktop providing at least one icon representing an application;

the framework receiving a gesture to activate the application;

the framework determining at least a portion of a composite display to render an active window associated with the application, wherein the composite display includes display areas for the two or more displays of the multi-screen device;

determining a display identifier for the active window;

determining a position in the window stack for the active window; and

generating a logic data structure for the active window to describe the position of the active window in the window stack, wherein the logic data structure includes the display identifier and the position in the window stack.

17. The method defined in claim 16, wherein the logic data structure further comprises a window identifier adapted to identify the active window in relation to other windows in the window stack.

18. The method defined in claim 16, wherein the window stack includes at least one of an inactive window and a desktop.

19. The method defined in claim 16, wherein the active window is also associated with an identifier for the composite display of the multi-screen device.

20. The method defined in claim 19, further comprising the framework rendering the composite display, wherein the composite display includes at least the active window.

ABSTRACT

Systems and methods are provides for creating a window stack for a multi-screen device. The stack is an arrangement of an active window and at least one other active or inactive window for at least one of the two displays. The system can receive activation of a window in the device. Upon activation of the window, the system can determine a display associated with the active window and can determine a position in the window stack for the active window. Then, the system can generate a logic data structure for the active window to describe the position of the active window in the window stack.

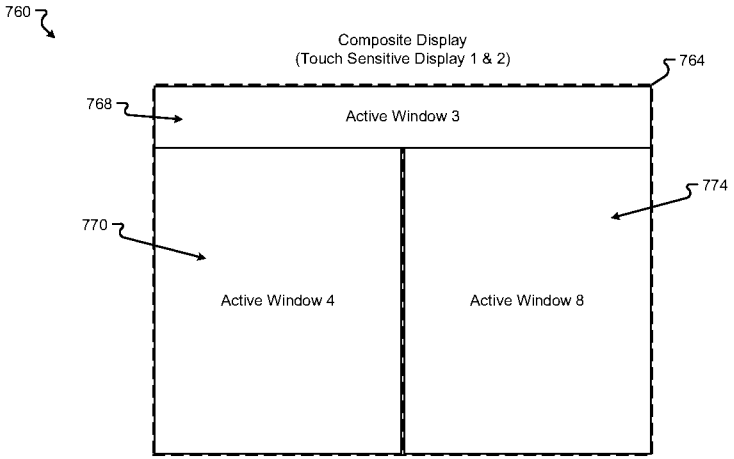


FIG. 7C

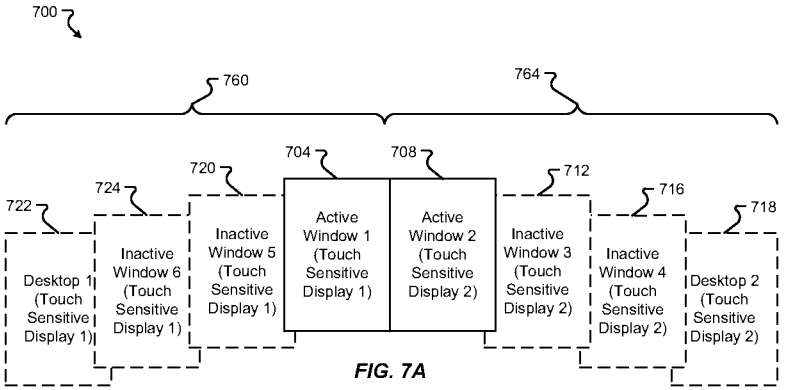


FIG. 7A

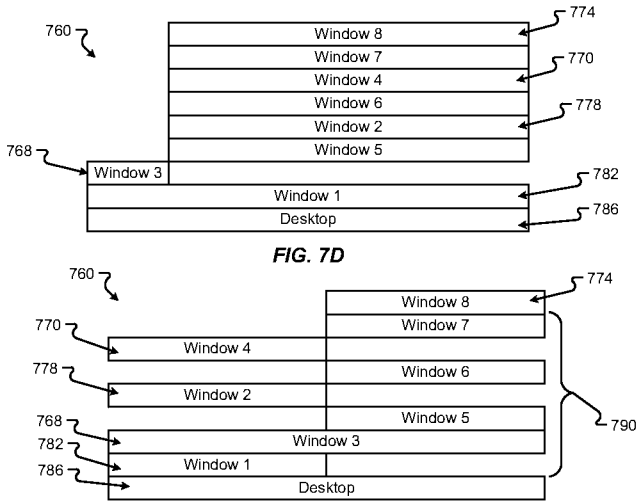


FIG. 7D

FIG. 7E

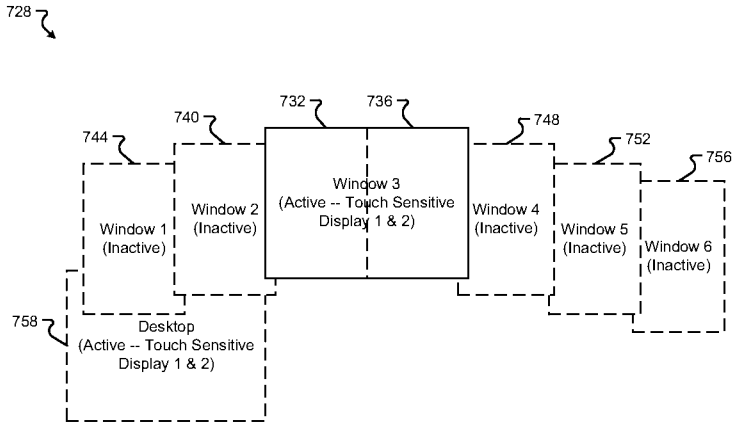


FIG. 7B

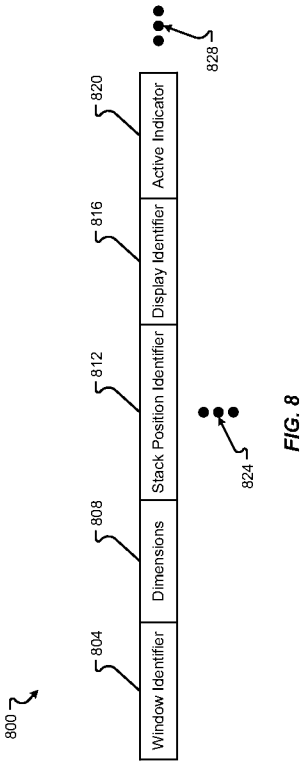

800
804
Window Identifier
808
Dimensions
812
Stack Position Identifier
816
Display Identifier
820
Active Indicator
824
828

FIG. 8

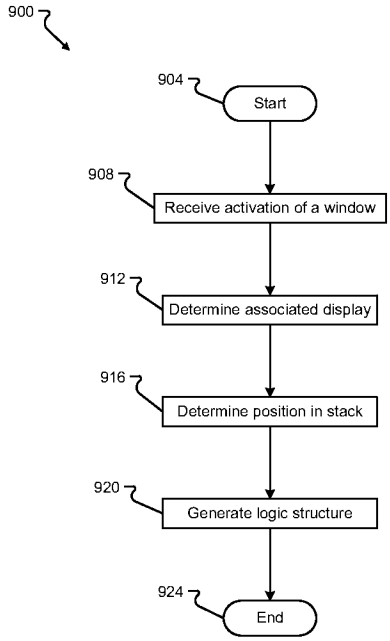

900
904
Start
908
Receive activation of a window
912
Determine associated display
916
Determine position in stack
920
Generate logic structure
924
End

FIG. 9

[0100] The device 100 manages desktops and/or windows with at least one window stack 700, as shown in Figs. 7A and 7B. A window stack 700 is a logical arrangement of active and/or inactive windows or display objects for a multi-screen device. For example, the window stack 700 may be logically similar to a deck of cards or a stack of bricks, where one or more windows or display objects (*e.g.*, desktops) are arranged in an order, as shown in Figs. 7A and 7B. An active window is a window that is currently being displayed on at least one of the touch sensitive displays 110, 114. For example, window 1 708 is an active windows and displayed on only one of the touch sensitive displays 114. In the embodiment shown in Fig. 7A, the device 100 is in a closed state 304 displaying window 1 708 in configuration 608. An inactive window is a window that was opened and displayed but is now "behind" an active window and not being displayed. In embodiments, an inactive window may be for an application that is suspended, and thus, the window is not displaying active content. For example, window 2 712 and window 3 716 are inactive windows.

[0101] A window stack 700 may have various arrangements or organizational structures. In the embodiment shown in Fig. 7A, the device 100 includes a first stack 704 associated with a first touch sensitive display 114, while the device 100 is in the closed state 304 in the second portrait configuration 608. Thus, each touch sensitive display 114 can have an associated window stack 704. In other embodiments, there is a single window stack that encompasses all of a composite display. The composite display is a logical structure that defines the entire display space including the two touch sensitive displays 110, 114. The device 100 can have a single stack for the composite display where the windows or display objects are sized to occupy a portion or all of the composite display. Thus, the stack 704 can represent a portion of a larger composite window stack, part of which is not shown because the device 100 is in a closed state 304.

[0102] In Fig. 7B, the two window stacks (or two portions of the window stack) 704, 724 may have a different number of windows or display objects arranged in the respective stacks 704, 724. Further, the two window stacks 704, 724 can also be identified differently and managed separately. As shown in Fig. 7A, the first window stack 704 can be arranged in order from a first window 708 to a next window 712 to a last window 716 and finally to a desktop 720, which, in embodiments, is at the "bottom" of the window stack 704. In embodiments, the desktop 720 is not always at the "bottom" as application windows can be arranged in the window stack below

the desktop 720, and the desktop 720 can be brought to the "top" of a stack over other windows during a desktop or other orientation change. For example, as shown in Fig. 7B, the device 100 transitions to an open state 320. The device 100 transitions to a display configuration as shown in Fig. 6C. As such, the touch sensitive display 110 has no windows associated with the touch sensitive display 110. Therefore, the desktop 720 is displayed for touch sensitive display 110. Thus, the second stack 724 can includes on the desktop 720, which, in embodiments, is a single desktop area, with desktop 720 under all the windows in both window stack 704 and window stack 724. A logical data structure for managing the two window stacks or single window stack having two portions 704, 724 may be as described in conjunction with Fig. 8.

[0103] The arrangement of the window stack 704 after the device is opened is shown in Figs. 7C through 7E. The window stack 704 is shown in three "elevation" views. In Fig. 7C, the top of the window stack 704/724 is shown. Two adjacent sides of the window stack 704/724 are shown in Fig. 7D and 7E. In this embodiment, the window stack 704/724 resembles a stack of bricks. The windows are stacked on each other. Looking from the top of the window stack 704/724 in Fig, 7C, only the top most windows in the window stack 704/724 are seen in different portions of a composite display 728. A desktop 786 or a window can occupy part or all of the composite display 728. Fig. C shows a composite display 728. The composite display 728 encompasses or includes the display area of all the touch sensitive displays 110,114. The size of the composite display 728 can change based on the orientation of the device 100. For example, the size of the composite display 728 of the device 100 when in the closed state may only include the area for one of the touch sensitive displays 110 or 114, as shown in Fig. 7A. The composite display 728 expands to include both touch sensitive displays 110,114 when the device 100 is opened, as shown in Fig. 7B and 7C. Some windows or display objects are associated with the composite display 728 and also change dimensions when the composite display 728 changes dimensions. In embodiments, one such display object may be the desktop 720, which can change dimensions upon a device 100 open to fill the composite display 728.

[0104] In the embodiment shown, the desktop 720 is the lowest display object, window, or "brick" in the window stack 704/724. Thereupon, window 1 708, window 2 712, and window 3 716 are layered. Window 1 708, window 2 712, and window 3 716 only occupy a portion of the composite display 728. Thus, another part of the stack 724 includes only the desktop 718. Only the top window or display object in any portion of the composite display 728 is actually rendered

and displayed. Thus, as shown in the top view in Fig. 7C, window 1 708 and the desktop 718 are displayed by being at the top of the different portions of the window stack 704/724. A window can be dimensioned to occupy only a portion of the composite display 728 to "reveal" windows lower in the window stack 704. For example, the desktop 718 is lower in the stack that window 1 708, window 2 712, and window 3 716 but is still displayed. This arrangement of windows and the desktop occurs when the device is opened when displaying top window in the window stack 704.

[0105] Where and how the desktop 718 is positioned within the stack can be a function of the orientation of the device 100, the context of what programs, functions, software, etc. are being executed on the device 100, how the stack is positioned when the device 100 is opened, etc. A logical data structure associated with the desktop 718 or the other windows may not change when the device 100 opens, but the logical data structures can determine how to display the windows and the desktop. When user interface or other events or tasks change the arrangement of the stack, the logical structures of the windows or desktop(s) can be changed to reflect the change in arrangement.

[0106] Another embodiment of a window stack arrangement 724, which has changed because a device has opened from a different configuration from Figs. 7A-7C, is shown in Figs. 7F through 7J. In the embodiment shown in Fig. 7F, the device 100 is in a closed state 304 displaying window 1 708 in configuration 608. Window 2 712 and window 3 716 are inactive windows. In the embodiment shown in Fig. 7F, the device 100 includes a second stack 724 associated with a second touch sensitive display 110, while the device 100 is in the closed state 304 in first portrait configuration 604. Thus, the touch sensitive display 110 can have an associated window stack 724. In other embodiments, there is a single window stack that encompasses all of a composite display. The device 100 can have a single stack for the composite display where the windows or display objects are sized to occupy a portion or all of the composite display. Thus, the stack 724 can represent a portion of a larger composite display stack, part of which is not shown because the device 100 is in a closed state 304.

[0107] In Fig. 7G, the two window stacks (or two portions of the window stack) 704, 724 may have a different number of windows or display objects arranged in the respective stacks 704, 724. Further, the two window stacks 704, 724 can also be identified differently and managed separately. As shown in Fig. 7F, the second window stack 724 can be arranged in order from a

first window 708 to a next window 712 to a last window 716 and finally to a desktop 720, which, in embodiments, is at the "bottom" of the window stack 704. In embodiments, the desktop 720 is not always at the "bottom" as application windows can be arranged in the window stack below the desktop 720, and the desktop 720 can be brought to the "top" of a stack over other windows, such as during an orientation change. For example, as shown in Fig. 7G, the device 100 transitions to an open state 320. The device 100 transitions to a display configuration as shown in Fig. 6C. As such, the touch sensitive display 114 has no windows associated with the touch sensitive display 114. Therefore, the desktop 720 is displayed for touch sensitive display 114. Thus, the second stack 704 can include the desktop 720, which, in embodiments, is a single desktop area, with desktop 720 under all the windows in both window stack 704 and window stack 724. A logical data structure for managing the two window stacks or single window stack, having two portions 704, 724, may be as described in conjunction with Fig. 8.

[0108] The arrangement of the window stack 724 after the device is opened is shown in Figs. 7H through 7J. The window stack 704 is shown in three "elevation" views. In Fig. 7H, the top of the window stack 704/724 is shown. Two adjacent "sides" of the window stack 704/724 are shown in Fig. 7I and 7J. In this embodiment, the window stack 704/724 resembles a stack of bricks. The windows are stacked on each other. Looking from the top of the window stack 704/724 in Fig, 7H, only the top most windows/display objects in the window stack 704/724 are seen in different portions of a composite display 728. A desktop 720 or a window can occupy part or all of the composite display 728.

[0109] In the embodiment shown, the desktop 720 is the lowest display object, window, or "brick" in the window stack 704/724. Thereupon, window 1 708, window 2 712, and window 3 716 are layered. Window 1 708, window 2 712, and window 3 716 only occupy a portion of the composite display 728. Thus, another part of the stack 724 includes only the desktop 718. Only the top window or display object in any portion of the composite display 728 is actually rendered and displayed. Thus, as shown in the top view in Fig. 7H, window 1 708 and the desktop 720 are displayed by being at the top of the different portions of the window stack 704/724. A window can be dimensioned to occupy only a portion of the composite display 728 to "reveal" windows lower in the window stack 704. For example, the desktop 718 is lower in the stack that window 1 708, window 2 712, and window 3 716 but is still displayed. This arrangement of windows

and the desktop occurs when the device is opened when displaying a top window in the window stack 704.

[0110] Where and how the desktop 718 is positioned within the stack can be a function of the orientation of the device 100, the context of what programs, functions, software, etc. are being executed on the device 100, how the stack is positioned when the device 100 is opened, etc. A logical data structure associated with the desktop 718 or the other windows may not change when the device 100 opens, but the logical data structures can determine how to display the windows and the desktop. When user interface or other events or tasks change the arrangement of the stack, the logical structures of the windows or desktop(s) can be changed to reflect the change in arrangement.

[0111] A logical data structure 800 for managing the arrangement of windows or desktops in a window stack is shown in Fig. 8. The logical data structure 800 can be any data structure used to store data whether an object, record, file, etc. The logical data structure 800 can be stored in any type of database or data storage system, regardless of protocol or standard. In embodiments, the logical data structure 800 includes one or more portions, fields, attributes, etc. that store data in a logical arrangement that allows for easy storage and retrieval of the information. Hereinafrer, these one or more portions, fields, attributes, etc. shall be described simply as fields. The fields can store data for a window identifier 804, dimensions 808, a stack position identifier 812, a display identifier 816, and/or an active indicator 820. Each window in a window stack can have an associated logical data structure 800. While only a single logical data structure 800 is shown in Fig. 8, there may be more or fewer logical data structures 800 used with a window stack (based on the number of windows or desktops in the stack), as represented by ellipses 824. Further, there may be more or fewer fields than those shown in Fig. 8, as represented by ellipses 828.

[0112] A window identifier 804 can include any identifier (ID) that uniquely identifies the associated window or display object in relation to other windows or display objects in the window stack. The window identifier 804 can be a globally unique identifier (GUID), a numeric ID, an alphanumeric ID, or other type of identifier. In embodiments, the window identifier 804 can be one, two, or any number of digits based on the number of windows or display objects that can be opened. In alternative embodiments, the size of the window identifier 804 may change

based on the number of windows or display objects opened. While the window or display object is open, the window identifier 804 may be static and remain unchanged.

[0113] Dimensions 808 can include dimensions for a window or display object in the composite display 704. For example, the dimensions 808 can include coordinates for two or more corners of the window or display object or may include one coordinate and dimensions for the width and height of the window or display object. These dimensions 808 can delineate what portion of the composite display 704 the window or display object may occupy, which may the entire composite display 704 or only part of composite display 704. For example, window 1 708 may have dimensions 880 that indicate that the window 1 708 will occupy only part of the display area for composite display 728, as shown in Figs. 7C and 7H. As windows or display objects are moved or inserted in the window stack, the dimensions 808 may change.

[0114] A stack position identifier 812 can be any identifier that can identify the position in the stack for the window or display object or may be inferred from the window's control record within a data structure, such as a list or a stack. The stack position identifier 812 can be a GUID, a numeric ID, an alphanumeric ID, or other type of identifier. Each window or display object can include a stack position identifier 812. For example, as shown in Fig. 7A, window 1 708 in stack 1 704 can have a stack position identifier 812 of 1 identifying that window 708 is the first window in the stack 704 and the active window. Similarly, window 2 712 can have a stack position identifier 812 of 2 representing that window 2 712 is the second window in the stack 704. Thus, depending on the type of stack, the stack position identifier 812 can represent a window's or display object's location in the stack.

[0115] A display identifier 816 can identify that the window or display object is associated with a particular display, such as the first display 110, the second display 114, or the composite display 728 composed of both displays. While this display identifier 816 may not be needed for a multi-stack system, as shown in Fig. 7A, the display identifier 816 can indicate whether a window or display object in the serial stack of Fig. 7B is displayed on a particular display. Thus, the desktop 720 may have two portions shown in Fig. 7C. The first portion may have a display identifier 816 for the first display 110 while the second portion may have a display identifier 816 for the second display 114. However, in alternative embodiments, the desktop 720 may have a single display identifier 816 identifying the composite display 728.

[0116] Similar to the display identifier 816, an active indicator 820 may not be needed with the dual stack system of Fig. 7A, as the window or display object in stack position 1 is active and displayed. In alternative embodiments, the active indicator 820 can indicate which window(s) in the stack is being displayed. Thus, window 1 708 may be shown and have an active indicator 820. The active indicator 820 can be a simple flag or bit that represents that the window or display object is active or displayed.

[0117] An embodiment of a method 900 for changing a window stack is shown in Fig. 9. While a general order for the steps of the method 900 is shown in Fig. 9. Generally, the method 900 starts with a start operation 904 and ends with an end operation 924. The method 900 can include more or fewer steps or can arrange the order of the steps differently than those shown in Fig. 9. The method 900 can be executed as a set of computer-executable instructions executed by a computer system and encoded or stored on a computer readable medium. Hereinafter, the method 900 shall be explained with reference to the systems, components, modules, software, data structures, user interfaces, etc. described in conjunction with Figs. 1-8.

[0118] A multi-screen device 100 can receive an orientation change, as described in Figs. 3A-3B, which changes the device 100 from a closed state 304 to an open state 320, in step 908. The orientation change can be detected and signaled by a hardware input from Hall Effect sensors, a timer, etc. The orientation change can be received by the Task Management Module 540 and sent to the Multi-Display Management Module 524. The Multi-Display Management Module 524 may interpret the change to changes the configuration of the display from a closed portrait display 604, 608 to an open portrait display (as shown in Fig. 6C) or from a landscape closed display 612, 616 to an open landscape configuration (as shown in Fig. 6F, as described in conjunction with Figs. 6A through 6F. In embodiments, the Task Management Module 540 places the user interface interaction in the task stack 552 to be acted upon by the Multi-Display Management Module 524. Further, the Task Management Module 540 waits for information from the Multi-Display Management Module 524 to send instructions to the Window Management Module 532 to create the window in the window stack 704.

[0119] The Multi-Display Management Module 524, upon receiving instruction from the Task Management Module 540, determines whether to reveal the desktop 786, in step 912. In embodiments, the desktop 786 may be at the bottom of the window stack 704/724 before the device 100 is opened. However, the device 100 may be displaying the last window in the stack

704/724. In other words, the top window in the stack 704/724 is being displayed and there are no windows in a position to fill the newly exposed display in the open state. For example, in Fig. 7A, there are no windows to the "left" of window 1 708. As such, the Multi-Display Management Module 524 needs to present the desktop 720 in display 110, as shown in Fig. 7B. As the desktop 720 generally is spread across the composite display 728, the desktop 720 is always display on a device 100 open unless another window covers the desktop 720. As there are no windows covering the desktop 720, as shown in Figs. 7B and 7F, the Multi-Display Management Module 524 determines to display the desktop 720 in the opened display.

[0120] In embodiments, the device state module 574 of the Multi-Display Management Module 524 may determine how the device is oriented or in what state the device is in, e.g., open, closed, portrait, etc. Further, the preferences module 572 and/or requirements module 580 may determine how the desktop 786 is to be displayed based on the preferences of the desktop 786. The Display Configuration Module 568 may then use the input from the device state module 574, preferences module 572, and/or other framework components 520 to evaluate the current window stack 704/724. The desktop 720 being shown in the new display typically does not affect other windows in the window stack 704/724 because the other windows do not move. However, the dimensions 808 of the desktop 720 can change as the desktop 720 is modified to fill the composite display 728, which expanded upon the opening of the device 100.

[0121] The visibility algorithm, in embodiments, determines for all portions of the composite display 728, which windows/display objects are at the top of the stack 704/724. For example, the visibility algorithm determines that, after the device 100 is opened, the desktop 786 is revealed in one portion of the stack 704, as viewed in Figs. 7B or 7F. Further, window 1 708 is displayed in the other portion of the stack 704, as viewed in Figs. 7B or 7F. Upon determining where to reveal the desktop 720, the Display Configuration Module 568 can change the display dimensions 808, display identifiers 816, and/or stack position identifiers 812 for the desktop 720. The Multi-Display Management Module 524 can then send the dimensions 808, display identifiers 816, and/or stack position identifiers 812 back to the Task Management Module 540.

[0122] In embodiments, the Task Management Module 540 sends the dimensions 808, display identifiers 816, and/or stack position identifiers 812, and/or other information and instructions to render the desktop 786 to the Window Management Module 532. The Window Management Module 532 and the Task Management Module 540 can change the logical data structure(s) 800,

in step 920. Both the Task Management Module 540 and the Window Management Module 532 may manage copies of the window stack 704/724. These copies of the window stack 704/724 can be synchronized or kept similar through communications between the Window Management Module 532 and the Task Management Module 540. Thus, the Window Management Module 532 and the Task Management Module 540, based on the information determined by the Multi-Display Management Module 524, can change dimensions 808, display identifiers 816, and/or stack position identifiers 812 for the desktop 720 and the one or more window(s). The logical data structure 800 may then be stored by both the Window Management Module 532 and the Task Management Module 540. Further, the Window Management Module 532 and the Task Management Module 540 may thereinafter manage the window stack 704/724and the logical data structure(s) 800.

[0123] The exemplary systems and methods of this disclosure have been described in relation to a window stack associated with a multi-screen device. However, to avoid unnecessarily obscuring the present disclosure, the preceding description omits a number of known structures and devices. This omission is not to be construed as a limitation of the scopes of the claims. Specific details are set forth to provide an understanding of the present disclosure. It should however be appreciated that the present disclosure may be practiced in a variety of ways beyond the specific detail set forth herein.

[0124] Furthermore, while the exemplary aspects, embodiments, and/or configurations illustrated herein show the various components of the system collocated, certain components of the system can be located remotely, at distant portions of a distributed network, such as a LAN and/or the Internet, or within a dedicated system. Thus, it should be appreciated, that the components of the system can be combined in to one or more devices, such as the software modules of Figs. 5A and 5B, or collocated on a particular node of a distributed network, such as an analog and/or digital telecommunications network, a packet-switch network, or a circuit-switched network. It will be appreciated from the preceding description, and for reasons of computational efficiency, that the components of the system can be arranged at any location within a distributed network of components without affecting the operation of the system. For example, the various components can be located in a switch such as a PBX and media server, gateway, in one or more communications devices, at one or more users' premises, or some

combination thereof. Similarly, one or more functional portions of the system could be distributed between a telecommunications device(s) and an associated computing device.

[0125] Furthermore, it should be appreciated that the various links connecting the elements can be wired or wireless links, or any combination thereof, or any other known or later developed element(s) that is capable of supplying and/or communicating data to and ftom the connected elements. These wired or wireless links can also be secure links and may be capable of communicating encrypted information. Transmission media used as links, for example, can be any suitable carrier for electrical signals, including coaxial cables, copper wire and fiber optics, and may take the form of acoustic or light waves, such as those generated during radio-wave and infra-red data communications.

[0126] Also, while the flowcharts have been discussed and illustrated in relation to a particular sequence of events, it should be appreciated that changes, additions, and omissions to this sequence can occur without materially affecting the operation of the disclosed embodiments, configuration, and aspects.

[0127] A number of variations and modifications of the disclosure can be used. It would be possible to provide for some features of the disclosure without providing others. For example in one alternative embodiment, the window stack 700 resembles a carousel or rolodex rather than a deck of cards. Thus, windows may circulate ftom one touch sensitive display 110 to another touch sensitive display 114. For example, window may be push to the right and end up at the end of stack behind another window. If the stack continues to move to the right, eventually the window would appear on the second touch sensitive display even though the window was opened in the first touch sensitive display 110. These movements and changes in the stacks may be managed using the methods and logical data structures discussed above. In another alternative embodiment, there may be other arrangement for the window stack.

[0128] In yet another embodiment, the systems and methods of this disclosure can be implemented in conjunction with a special purpose computer, a programmed microprocessor or microcontroller and peripheral integrated circuit element(s), an ASIC or other integrated circuit, a digital signal processor, a hard-wired electronic or logic circuit such as discrete element circuit, a programmable logic device or gate array such as PLD, PLA, FPGA, PAL, special purpose computer, any comparable means, or the like. In general, any device(s) or means capable of implementing the methodology illustrated herein can be used to implement the various aspects of

this disclosure. Exemplary hardware that can be used for the disclosed embodiments, configurations and aspects includes computers, handheld devices, telephones (e.g., cellular, Internet enabled, digital, analog, hybrids, and others), and other hardware known in the art. Some of these devices include processors (e.g., a single or multiple microprocessors), memory, nonvolatile storage, input devices, and output devices. Furthermore, alternative software implementations including, but not limited to, distributed processing or component/object distributed processing, parallel processing, or virtual machine processing can also be constructed to implement the methods described herein.

[0129] In yet another embodiment, the disclosed methods may be readily implemented in conjunction with software using object or object-oriented software development environments that provide portable source code that can be used on a variety of computer or workstation platforms. Alternatively, the disclosed system may be implemented partially or fully in hardware using standard logic circuits or VLSI design. Whether software or hardware is used to implement the systems in accordance with this disclosure is dependent on the speed and/or efficiency requirements of the system, the particular function, and the particular software or hardware systems or microprocessor or microcomputer systems being utilized.

[0130] In yet another embodiment, the disclosed methods may be partially implemented in software that can be stored on a storage medium, executed on programmed general-purpose computer with the cooperation of a controller and memory, a special purpose computer, a microprocessor, or the like. In these instances, the systems and methods of this disclosure can be implemented as program embedded on personal computer such as an applet, JAVA® or CGI script, as a resource residing on a server or computer workstation, as a routine embedded in a dedicated measurement system, system component, or the like. The system can also be implemented by physically incorporating the system and/or method into a software and/or hardware system.

[0131] Although the present disclosure describes components and functions implemented in the aspects, embodiments, and/or configurations with reference to particular standards and protocols, the aspects, embodiments, and/or configurations are not limited to such standards and protocols. Other similar standards and protocols not mentioned herein are in existence and are considered to be included in the present disclosure. Moreover, the standards and protocols mentioned herein and other similar standards and protocols not mentioned herein are periodically

superseded by faster or more effective equivalents having essentially the same functions. Such replacement standards and protocols having the same functions are considered equivalents included in the present disclosure.

[0132] The present disclosure, in various aspects, embodiments, and/or configurations, includes components, methods, processes, systems and/or apparatus substantially as depicted and described herein, including various aspects, embodiments, configurations embodiments, subcombinations, and/or subsets thereof. Those of skill in the art will understand how to make and use the disclosed aspects, embodiments, and/or configurations after understanding the present disclosure. The present disclosure, in various aspects, embodiments, and/or configurations, includes providing devices and processes in the absence of items not depicted and/or described herein or in various aspects, embodiments, and/or configurations hereof, including in the absence of such items as may have been used in previous devices or processes, e.g., for improving performance, achieving ease and\or reducing cost of implementation.

[0133] The foregoing discussion has been presented for purposes of illustration and description. The foregoing is not intended to limit the disclosure to the form or forms disclosed herein. In the foregoing Detailed Description for example, various features of the disclosure are grouped together in one or more aspects, embodiments, and/or configurations for the purpose of streamlining the disclosure. The features of the aspects, embodiments, and/or configurations of the disclosure may be combined in alternate aspects, embodiments, and/or configurations other than those discussed above. This method of disclosure is not to be interpreted as reflecting an intention that the claims require more features than are expressly recited in each claim. Rather, as the following claims reflect, inventive aspects lie in less than all features of a single foregoing disclosed aspect, embodiment, and/or configuration. Thus, the following claims are hereby incorporated into this Detailed Description, with each claim standing on its own as a separate preferred embodiment of the disclosure.

[0134] Moreover, though the description has included description of one or more aspects, embodiments, and/or configurations and certain variations and modifications, other variations, combinations, and modifications are within the scope of the disclosure, e.g., as may be within the skill and knowledge of those in the art, after understanding the present disclosure. It is intended to obtain rights which include alternative aspects, embodiments, and/or configurations to the extent permitted, including alternate, interchangeable and/or equivalent structures, functions,

ranges or steps to those claimed, whether or not such alternate, interchangeable and/or equivalent structures, functions, ranges or steps are disclosed herein, and without intending to publicly dedicate any patentable subject matter.

What is claimed is:

1. A computer readable medium, having stored thereon, computer-executable instructions executable by a processor, the computer-executable instructions causing the processor to execute a method for managing a window stack for a device, the computer-executable instructions comprising:

instructions to receive an orientation change, wherein the orientation change causes the device to transition from a closed state to an open state, wherein, in the closed state, a first display is active and, in the open state, both the first display and a second display are active;

instructions to determine if a desktop should be displayed on the second display; and

afrer determining the desktop should be displayed on the second display, instructions to display the desktop on the second display and a first window on the first display.

2. The computer readable medium as defined in claim 1, wherein the window stack is a logical arrangement of active and inactive windows for the multi-screen device.

3. The computer readable medium as defined in claim 2, wherein the window stack provides a representation of windows or display objects to the user.

4. The computer readable medium as defined in claim 3, wherein the desktop is a display object.

5. The computer readable medium as defined in claim 4, wherein the desktop is displayed across a composite display, wherein the composite display occupies substantially all of the at least the two or more touch sensitive displays, and wherein the first display and second display are portions of the composited display.

6. The computer readable medium as defined in claim 5, further comprising instructions to modify a logical data structure associated with the desktop.

7. The computer readable medium as defined in claim 6, wherein the logical data structure comprises one or more of:

a window identifier adapted to identify the desktop in relation to other windows in the window stack;

a window stack position identifier adapted to identify the position in the window stack for the desktop; and

a display identifier adapted to identify the composite display of the multi-screen device, which the desktop is associated.

8. The computer readable medium as defined in claim 6, wherein a first display of the multi-screen device is associated with a first portion of the window stack and a second display of the multi-screen device is associated with a second portion of the window stack.

9. The computer readable medium as defined in claim 8, wherein the logic data structure further comprises a window stack identifier adapted to identify both the first portion and second portion of which the desktop is associated.

10. A device comprising:

at least two displays;

a memory;

a processor in communication with the memory and each of the at least two displays, the processor operable to:

provide a composite display, wherein the composite display comprises one touch sensitive displays in a closed state;

receive an orientation change in the multi-screen device, wherein the orientation change causes the multi-screen device to transition from a closed state to an open state, wherein, in the open state, the composite display comprises a first portion associated with the first touch sensitive display and a second portion associated with a second touch sensitive display;

expand a desktop to cover the composite display;

determine a first window is displayed in a first portion of the composite display and the desktop is displayed in a second portion of the composite display; and

display the desktop on the second touch sensitive display and the first window on the first touch sensitive display.

11. The device as defined in claim 10, wherein the desktop is altered upon the transition to the open state.

12. The device as defined in claim 11, further comprising altering a logic data structure associated with the desktop.

13. The device as defined in claim 12, wherein the desktop is associated with both of the two touch sensitive displays.

14. The device as defined in claim 13, wherein the logic data structure comprises one or more of:

a window identifier adapted to identify the desktop in relation to other windows in the window stack;

a window stack position identifier adapted to identify the top position in the window stack for the desktop in at least the second portion of the composited display; and

at least two display identifiers adapted to identify each of the at least two touch sensitive displays of the multi-screen device which the desktop is associated.

15. The device as defined in claim 14, wherein the desktop is at the bottom of the opened at a bottom of a window stack and is displayed because no window covers the desktop in at least a portion of the composite display.

16. The device as defined in claim 15, wherein the first window covers at least a portion of the desktop.

17. A method for presenting a display for a multi-screen device, the method comprising:

providing a composite display that spans at least a portion of a first touch sensitive display when the multi-screen device is in a closed state;

displaying a first window at a top of a window stack when the device is in the closed state;

receiving an orientation change for the multi-screen device, wherein the orientation change is a transition from the closed state to an open state;

changing the composite display to span at least a portion of the first touch sensitive display and a second touch sensitive display of the multi-screen device, wherein a first portion of the composite display is associated with the first touch sensitive display and a second portion of the composite display is associated with the second touch sensitive display;

determining that a desktop is associated with the composite display;

modifying the desktop to expand over the composite display;

determining the first window is at the top of the window stack in the first portion of the composite display;

determining the desktop is at the top of the window stack in the second portion of the composite display;

displaying the first window on the first touch sensitive display; and

displaying the desktop on the second touch sensitive display.

18. The method defined in claim 17, wherein the desktop is at a bottom of the window stack.

19. The method defined in claim 18, wherein the first window is a last window in a window stack comprising two or more windows.

20. The method defined in claim 19, wherein the first touch sensitive display is part of a primary screen.

ABSTRACT

Systems and methods are provides for displaying a desktop for a multi-screen device in response to opening the device. The window stack can change based on the change in the orientation of the device. The system can receive an orientation change that transitions the device from a closed state to an open state. A previously created in the stack can expand over the area of the two or more displays comprising the device when opened. A desktop expands to fill the display area and be displayed on the second of the displays afrer the device is opened.

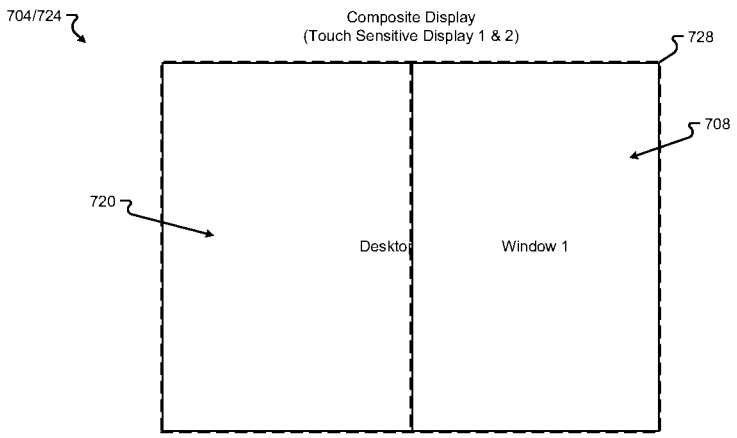


FIG. 7C

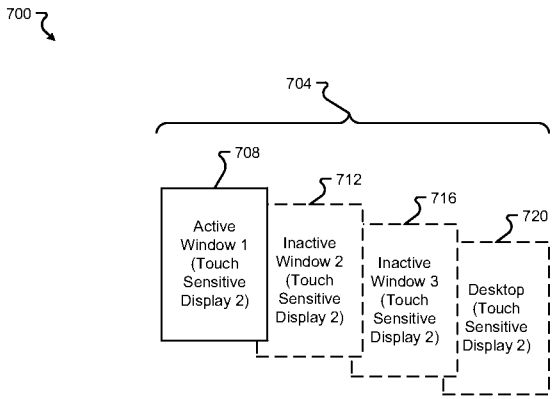


FIG. 7A

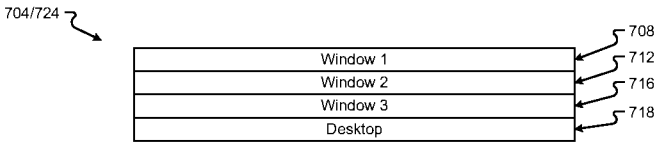


FIG. 7D

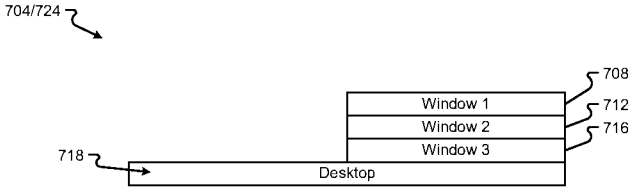


FIG. 7E

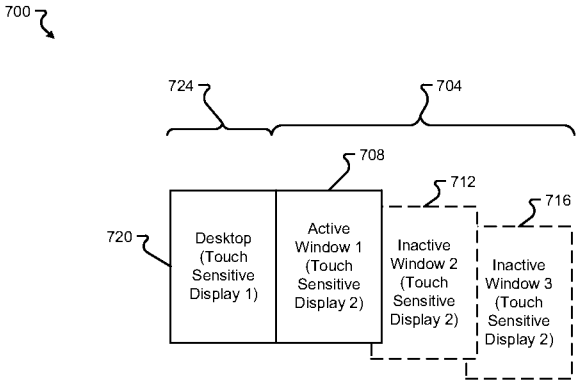


FIG. 7B

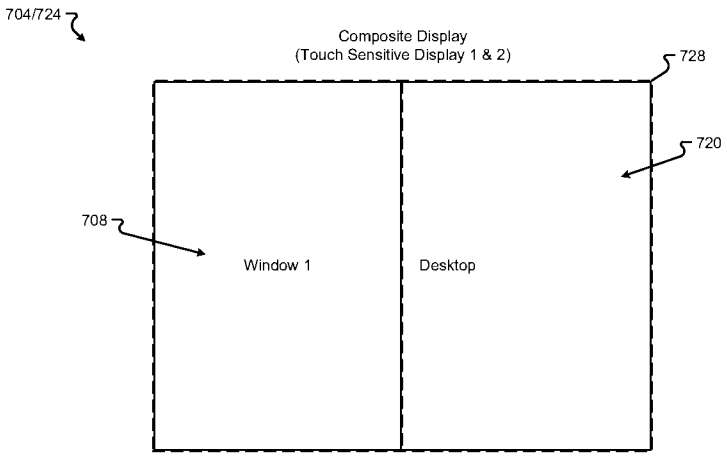


FIG. 7H

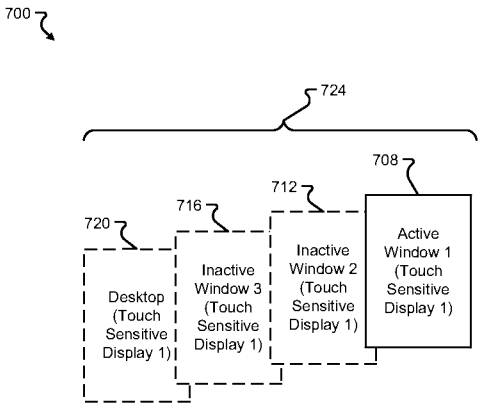


FIG. 7F

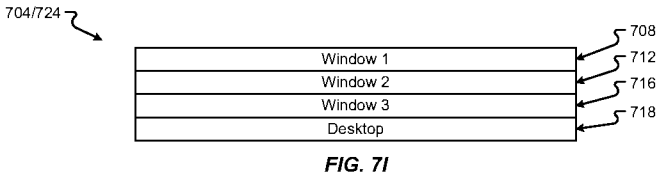


FIG. 7I

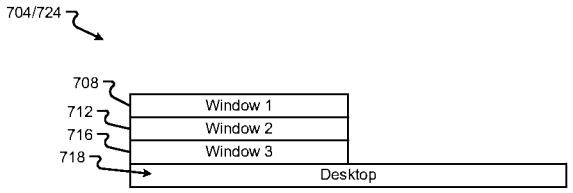


FIG. 7J

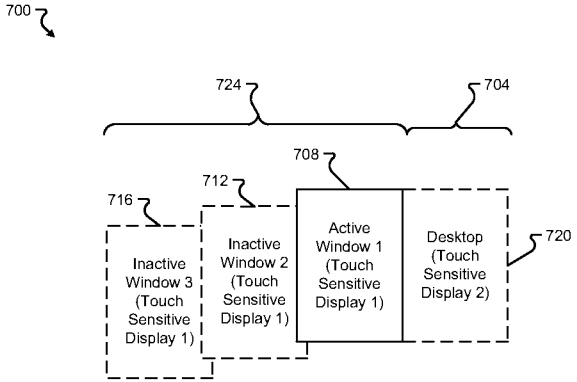


FIG. 7G

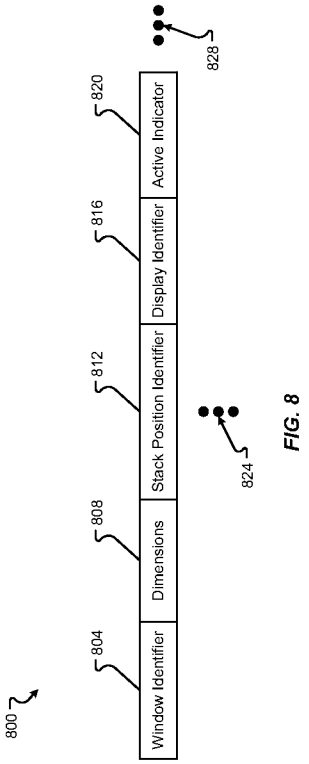

800
804
Window Identifier
808
Dimensions
812
Stack Position Identifier
816
Display Identifier
820
Active Indicator
824
828

FIG. 8

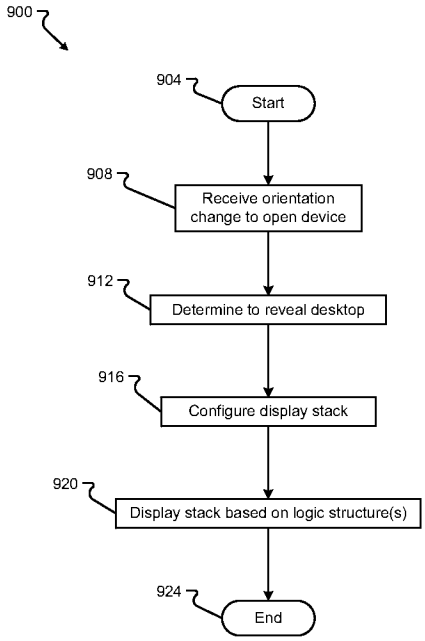

900
904
Start
908
Receive orientation change to open device
912
Determine to reveal desktop
916
Configure display stack
920
Display stack based on logic structure(s)
924
End

FIG. 9

[0100] The device 100 manages desktops and/or windows with at least one window stack 700, 728, as shown in Figs. 7A and 7B. A window stack 700, 728 is a logical arrangement of active and/or inactive windows for a multi-screen device. For example, the window stack 700, 728 may be logically similar to a deck of cards or a stack of bricks, where one or more windows or desktops are arranged in order, as shown in Figs. 7A and 7B. An active window is a window that is currently being displayed on at least one of the touch sensitive displays 110, 114. For example, windows 104 and 108 are active windows and are displayed on touch sensitive displays 110 and 114. An inactive window is a window that was opened and displayed but is now "behind" an active window and not being displayed. In embodiments, an inactive window may be for an application that is suspended, and thus, the window is not displaying active content. For example, windows 712, 716, 720, and 724 are inactive windows.

[0101] A window stack 700, 728 may have various arrangements or organizational structures. In the embodiment shown in Fig. 7A, the device 100 includes a first stack 760 associated with a first touch sensitive display 110 and a second stack associated with a second touch sensitive display 114. Thus, each touch sensitive display 110, 114 can have an associated window stack 760, 764. These two window stacks 760, 764 may have different numbers of windows arranged in the respective stacks 760, 764. Further, the two window stacks 760, 764 can also be identified differently and managed separately. Thus, the first window stack 760 can be arranged in order from a first window 704 to a next window 720 to a last window 724 and finally to a desktop 722, which, in embodiments, is at the "bottom" of the window stack 760. In embodiments, the desktop 722 is not always at the "bottom" as application windows can be arranged in the window stack below the desktop 722, and the desktop 722 can be brought to the "top" of a stack over other windows during a desktop reveal. Likewise, the second stack 764 can be arranged from a first window 708 to a next window 712 to a last window 716, and finally to a desktop 718, which, in embodiments, is a single desktop area, with desktop 722, under all the windows in both window stack 760 and window stack 764. A logical data structure for managing the two window stacks 760, 764 may be as described in conjunction with Fig. 8.

[0102] Another arrangement for a window stack 728 is shown in Fig. 7B. In this embodiment, there is a single window stack 728 for both touch sensitive displays 110, 114. Thus, the window stack 728 is arranged from a desktop 758 to a first window 744 to a last window 756. A window can be arranged in a position among all windows without an association to a specific touch

sensitive display 110, 114. In this embodiment, a window is in the order of windows. Further, at least one window is identified as being active. For example, a single window may be rendered in two portions 732 and 736 that are displayed on the first touch sensitive screen 110 and the second touch sensitive screen 114. The single window may only occupy a single position in the window stack 728 although it is displayed on both displays 110, 114.

[0103] Yet another arrangement of a window stack 760 is shown in Figs. 7C through 7E. The window stack 760 is shown in three "elevation" views. In Fig. 7C, the top of the window stack 760 is shown. Two sides of the window stack 760 are shown in Figs. 7D and 7E. In this embodiment, the window stack 760 resembles a stack of bricks. The windows are stacked on each other. Looking from the top of the window stack 760 in Fig, 7C, only the top most windows in the window stack 760 are seen in different portions of the composite display 764. The composite display 764 represents a logical model for the entire display area of the device 100, which can include touch sensitive display 110 and touch sensitive display 114. A desktop 786 or a window can occupy part or all of the composite display 764.

[0104] In the embodiment shown, the desktop 786 is the lowest display or "brick" in the window stack 760. Thereupon, window 1 782, window 2 782, window 3 768, and window 4 770 are layered. Window 1 782, window 3 768, window 2 782, and window 4 770 only occupy a portion of the composite display 764. Thus, another part of the stack 760 includes window 8 774 and windows 5 through 7 shown in section 790. Only the top window in any portion of the composite display 764 is actually rendered and displayed. Thus, as shown in the top view in Fig. 7C, window 4 770, window 8 774, and window 3 768 are displayed as being at the top of the display in different portions of the window stack 760. A window can be dimensioned to occupy only a portion of the composite display 760 to "reveal" windows lower in the window stack 760. For example, window 3 768 is lower in the stack than both window 4 770 and window 8 774 but is still displayed.

[0105] When a new window is moved, the moved window is generally positioned at the top of the stack if moved onto another display. If the window is moved off of the display, generally a window behind the moved window is brought to the top of the stack. Where and how the window is positioned within the stack can be a function of the orientation of the device 100, the context of what programs, functions, software, etc. are being executed on the device 100, how the stack is positioned when the new window is opened, etc. To move the window in the stack,

the position in the stack for the window is determined and the touch sensitive display 110, 114 to which the window is associated may also be determined. With this information, a logical data structure associated with the window can be changed. When user interface or other events or tasks change the arrangement of windows, the window stack(s) can be changed to reflect the change in arrangement. It should be noted that these same concepts described above can be used to manage the one or more desktops for the device 100.

[0106] An embodiment of a window stack arrangement 760 changed because of one or more a window moves is shown in Figs. 7F through 7H. The window stack 760 is shown in a next state (from Figs. 7C through 7E) in three views, Figs. 7F, 7G, and 7H. The top of the window stack 760 is shown in Fig. 7H, which is a view of what is displayed on the device 100. The two "sides" of the window stack 760 are shown in Figs. 7F and 7G. In this embodiment, the window 8 774 is moved by a gesture received from the touch sensitive display 110, 114 or from the gesture capture region 120, 124. In interpreting the gesture, the Task Management Module 540 can request the Multi-Display Management Module 524 to determine where to position window 8 774. The Multi-Display Management Module 524 may then change the dimensions of the window to position the window at the top of the stack 760 but displayed in the first touch sensitive display 110. Thus, looking from the top of the window stack 760 in Figs. 7C and 7H, window 8 774 moves from the right side of the composite display 764 to the lefr side of the composite display 764. Similarly, Window 8 774 moves from the right side of the stack 760 (shown in Fig. 7E) to the left side of the stack 760, as shown in Fig. 7G.

[0107] Based on the movement of window 8 774, desktop 786 is the lowest display or "brick" in the window stack 760. Thereupon, window 1 782, window 2 782, window 3 768, and window 4 770 are layered. Window 1 782, window 3 768, window 2 782, and window 4 770 only occupy a portion of the composite display 764. Thus, another part of the stack 760 includes window 8 774, window 5 794, and window 7 776. Only the top window in any portion of the composite display 764 is actually rendered and displayed. Thus, as shown in the top view in Fig. 7H, window 8 774 is still displayed but on the left side of the composite display 764. However, the movement of window 8 774 "uncovered" window 7 776, which is now displayed on the right side of the composite display 764. Further, window 3 768 is still displayed as being at the top of the display. These windows 774, 776, and 768 are all at different levels of the stack 760 and are in different portions of the window stack 760. A window can be dimensioned to occupy only a

portion of the composite display 760 to "reveal" windows lower in the window stack 760, which is how the re-dimensioning of window 8 774 uncovered window 7 776.

[0108] Still another embodiment of a window stack arrangement 760 that is changed because of one or more a window moves is shown in Figs. 7I through 7K. The window stack 760 is shown in a next state (from Figs. 7C through 7E) in three views, Figs. 7I, 7J, and 7K. The top of the window stack 760 is shown in Fig. 7I, which is a view of what is displayed on the device 100. The two "sides" of the window stack 760 are shown in Figs. 7J and 7K. In this embodiment, window 7 776 is moved by a gesture received from the touch sensitive display 110, 114 or from the gesture capture region 120, 124. In interpreting the gesture, the Task Management Module 540 can request the Multi-Display Management Module 524 to determine where to position window 7 776. The Multi-Display Management Module 524 may then change the dimensions of window 7 776 to position the window at the top of the stack 760 but displayed in the first touch sensitive display 110. Thus, looking from the top of the window stack 760 in Figs. 7H and 7I, window 7 776 moves from the right side of the composite display 764 to the left side of the composite display 764. Similarly, window 7 776 moves from the right side of the stack 760 (shown in Fig. 7G) to the left side of the stack 760, as shown in Fig. 7K.

[0109] Based on the movement of window 7 776, desktop 786 is the lowest display or "brick" in the window stack 760. Thereupon, window 1 782, window 2 782, window 3 768, and window 4 770 are layered. Window 1 782, window 3 768, window 2 782, and window 4 770 only occupy a portion of the composite display 764. Thus, another part of the stack 760 includes window 8 774, window 5 794, and window 7 776. Only the top window in any portion of the composite display 764 is actually rendered and displayed. Thus, as shown in the top view in Fig. 7I, window 7 776 is still displayed but on the left side of the composite display 764. However, the movement of window 7 776 "uncovered" window 6 780, which is now displayed on the right side of the composite display 764. Further, window 3 768 is still displayed as being at the top of the display. Window 8 774 is now covered by window 7 776. The displayed windows 776, 780, and 768 are all at different levels and are in different portions of the window stack 760. A window can be dimensioned to occupy only a portion of the composite display 760 to "reveal" windows lower in the window stack 760, which is how the re-dimensioning of window 7 776 uncovered window 6 780 and covered window 8 774. A logical data structure to manage the window stack can be as described in conjunction with Fig. 8.

[0110] A logical data structure 800 for managing the arrangement of windows or desktops in a window stack is shown in Fig. 8. The logical data structure 800 can be any data structure used to store data whether an object, record, file, etc. The logical data structure 800 can be stored in any type of database or data storage system, regardless of protocol or standard. In embodiments, the logical data structure 800 includes one or more portions, fields, attributes, etc. that store data in a logical arrangement that allows for easy storage and retrieval of the information. Hereinafter, these one or more portions, fields, attributes, etc. shall be described simply as fields. The fields can store data for a window identifier 804, dimensions 808, a stack position identifier 812, a display identifier 816, and/or an active indicator 820. Each window in a window stack can have an associated logical data structure 800. While only a single logical data structure 800 is shown in Fig. 8, there may be more or fewer logical data structures 800 used with a window stack (based on the number of windows or desktops in the stack), as represented by ellipses 824. Further, there may be more or fewer fields than those shown in Fig. 8, as represented by ellipses 828.

[0111] A window identifier 804 can include any identifier (ID) that uniquely identifies the associated window in relation to other windows in the window stack. The window identifier 804 can be a globally unique identifier (GUID), a numeric ID, an alphanumeric ID, or other type of identifier. In embodiments, the window identifier 804 can be one, two, or any number of digits based on the number of windows that can be opened. In alternative embodiments, the size of the window identifier 804 may change based on the number of windows opened. While the window is open, the window identifier 804 may be static and remain unchanged.

[0112] Dimensions 808 can include dimensions for a window in the composite display 760. For example, the dimensions 808 can include coordinates for two or more corners of the window or may include one coordinate and dimensions for the width and height of the window. These dimensions 808 can delineate what portion of the composite display 760 the window may occupy, which may the entire composite display 760 or only part of composite display 760. For example, window 4 770 may have dimensions 880 that indicate that the window 770 will occupy only part of the display area for composite display 760, as shown in Figs. 7c through 7E. As windows are moved or inserted in the window stack, the dimensions 808 may change.

[0113] A stack position identifier 812 can be any identifier that can identify the position in the stack for the window or may be inferred from the window's control record within a data

structure, such as a list or a stack. The stack position identifier 812 can be a GUID, a numeric ID, an alphanumeric ID, or other type of identifier. Each window or desktop can include a stack position identifier 812. For example, as shown in Fig. 7A, window 1 704 in stack 1 760 can have a stack position identifier 812 of 1 identifying that window 704 is the first window in the stack 760 and the active window. Similarly, window 6 724 can have a stack position identifier 812 of 3 representing that window 724 is the third window in the stack 760. Window 2 708 can also have a stack position identifier 812 of 1 representing that window 708 is the first window in the second stack 764. As shown in Fig. 7B, window 1 744 can have a stack position identifier 812 of 1, window 3, rendered in portions 732 and 736, can have a stack position identifier 812 of 3, and window 6 756 can have a stack position identifier 812 of 6. Thus, depending on the type of stack, the stack position identifier 812 can represent a window's location in the stack.

[0114] A display identifier 816 can identify that the window or desktop is associated with a particular display, such as the first display 110 or the second display 114, or the composite display 760 composed of both displays. While this display identifier 816 may not be needed for a multi-stack system, as shown in Fig. 7A, the display identifier 816 can indicate whether a window in the serial stack of Fig. 7B is displayed on a particular display. Thus, window 3 may have two portions 732 and 736 in Fig. 7B. The first portion 732 may have a display identifier 816 for the first display while the second portion 736 may have a display identifier 816 for the second display 114. However, in alternative embodiments, the window may have two display identifier 816 that represent that the window is displayed on both of the displays 110, 114, or a display identifier 816 identifying the composite display. In another alternate embodiment, the window may have a single display identifier 816 to represent that the window is displayed on both of the displays 110, 114.

[0115] Similar to the display identifier 816, an active indicator 820 may not be needed with the dual stack system of Fig. 7A, as the window in stack position 1 is active and displayed. In the system of Fig. 7B, the active indicator 820 can indicate which window(s) in the stack is being displayed. Thus, window 3 may have two portions 732 and 736 in Fig. 7. The first portion 732 may have an active indicator 820 while the second portion 736 may also have an active indicator 820. However, in alternative embodiments, window 3 may have a single active indicator 820. The active indicator 820 can be a simple flag or bit that represents that the window is active or displayed.

[0116] An embodiment of a method 900 for changing a window stack is shown in Fig. 9. While a general order for the steps of the method 900 is shown in Fig. 9. Generally, the method 900 starts with a start operation 904 and ends with an end operation 928. The method 900 can include more or fewer steps or can arrange the order of the steps differently than those shown in Fig. 9. The method 900 can be executed as a set of computer-executable instructions executed by a computer system and encoded or stored on a computer readable medium. Hereinafter, the method 900 shall be explained with reference to the systems, components, modules, software, data structures, user interfaces, etc. described in conjunction with Figs. 1-8.

[0117] A multi-screen device 100 can receive a gesture, as described in Figs. 4A-4H, directed to a window, in step 908. The gesture can be a hardware input from a touch sensitive display 110, 114, from a gesture capture region 120, 124, or some other user interface. The received gesture can be received by the Task Management Module 540 and sent to the Gesture Module 576 of the Multi-Display Management Module 524. The Gesture Module 576 may interpret the gesture as wanting to move a window, as described in conjunction with Figs. 7F through 7K. In embodiments, the Task Management Module 540 places the user interface interaction in the task stack 552 to be acted upon by the Multi-Display Management Module 524. Further, the Task Management Module 540 waits for information from the Multi-Display Management Module 524 to send instructions to the Window Management Module 532 to create the window in the window stack.

[0118] The Multi-Display Management Module 524, upon receiving instruction from the Task Management Module 540, determines to which touch portion of the composite display 760, the moved window should be associated, in step 912. For example, window 8 774 is associated with display 114, which is the right portion of the composite display 764, as shown in Fig. 7C. The movement of window 8 774, as described in conjunction with Figs. 7F through 7H, changes the association of window 8 to display 110, which is the left side of the composite display 764, as shown in Fig. 7H. In embodiments, the device state module 574 of the Multi-Display Management Module 524 may determine how the device is oriented or in what state the device is in, e.g., open, closed, portrait, etc. Further, the preferences module 572 and/or requirements module 580 may determine how the window is to be displayed. The gesture module 576 may determine the user's intentions about how the window is to be opened based on the type of gesture and the location of where the gesture is made.

[0119] The Display Configuration Module 568 may then use the input from the device state module 574, preferences module 572, and/or gesture module 576 to evaluate the current window stack 760 to determine the best place and the best dimensions, based on a visibility algorithm, to move the window. Thus, the Display Configuration Module 568 determines the best place to put the window at the top of the window stack 760, in step 916. The visibility algorithm, in embodiments, determines for all portions of the composite display, which windows are at the top of the stack. For example, the visibility algorithm determines that, after window 8 is moved, window 3 768, window 7 776, and window 8 774 are at the top of the stack 760, as viewed in Figs. 7F through 7H. Upon determining where to open the windows, the Display Configuration Module 568 can change the display identifiers 816, dimensions 808, and/or stack position identifiers 812 for the one or more windows affected by the movement. The Multi-Display Management Module 524 can then send the display identifiers 816, dimensions 808, and/or stack position identifiers 812 back to the Task Management Module 540.

[0120] In embodiments, the Task Management Module 540 sends the display identifiers 816, dimensions 808, stack position identifiers 812, and/or other information and instructions to render the window to the Window Management Module 532. The Window Management Module 532 and the Task Management Module 540 can change the logical data structure(s) 800, in step 924. Both the Task Management Module 540 and the Window Management Module 532 may manage copies of the window stack. These copies of the window stack can be synchronized or kept similar through communications between the Window Management Module 532 and the Task Management Module 540. Thus, the Window Management Module 532 and the Task Management Module 540, based on the information determined by the Multi-Display Management Module 524, can change display identifiers 816, dimensions 808, and/or stack position identifiers 812 for window 8 774, window 7 776, and/or window 4 770. The logical data structure 800 may then be stored by both the Window Management Module 532 and the Task Management Module 540. Further, the Window Management Module 532 and the Task Management Module 540 may thereinafter manage the window stack and the logical data structure(s) 800.

[0121] The exemplary systems and methods of this disclosure have been described in relation to a window stack associated with a multi-screen device. However, to avoid unnecessarily obscuring the present disclosure, the preceding description omits a number of known structures

and devices. This omission is not to be construed as a limitation of the scopes of the claims. Specific details are set forth to provide an understanding of the present disclosure. It should however be appreciated that the present disclosure may be practiced in a variety of ways beyond the specific detail set forth herein.

[0122] Furthermore, while the exemplary aspects, embodiments, and/or configurations illustrated herein show the various components of the system collocated, certain components of the system can be located remotely, at distant portions of a distributed network, such as a LAN and/or the Internet, or within a dedicated system. Thus, it should be appreciated, that the components of the system can be combined in to one or more devices, such as the software modules of Figs. 5A and 5B, or collocated on a particular node of a distributed network, such as an analog and/or digital telecommunications network, a packet-switch network, or a circuit-switched network. It will be appreciated from the preceding description, and for reasons of computational efficiency, that the components of the system can be arranged at any location within a distributed network of components without affecting the operation of the system. For example, the various components can be located in a switch such as a PBX and media server, gateway, in one or more communications devices, at one or more users' premises, or some combination thereof. Similarly, one or more functional portions of the system could be distributed between a telecommunications device(s) and an associated computing device.

[0123] Furthermore, it should be appreciated that the various links connecting the elements can be wired or wireless links, or any combination thereof, or any other known or later developed element(s) that is capable of supplying and/or communicating data to and from the connected elements. These wired or wireless links can also be secure links and may be capable of communicating encrypted information. Transmission media used as links, for example, can be any suitable carrier for electrical signals, including coaxial cables, copper wire and fiber optics, and may take the form of acoustic or light waves, such as those generated during radio-wave and infra-red data communications.

[0124] Also, while the flowcharts have been discussed and illustrated in relation to a particular sequence of events, it should be appreciated that changes, additions, and omissions to this sequence can occur without materially affecting the operation of the disclosed embodiments, configuration, and aspects.

[0125] A number of variations and modifications of the disclosure can be used. It would be possible to provide for some features of the disclosure without providing others. For example in one alternative embodiment, the window stack 700 resembles a carousel or rolodex rather than a deck of cards. Thus, windows may circulate from one touch sensitive display 110 to another touch sensitive display 114. For example, window 6 724 may be push to the right and end up at the end of stack 764 behind window 716. If the stack continues to move to the right, eventually window 6 724 would appear on the second touch sensitive display 114 even though the window 724 was opened in the first touch sensitive display 110. These movements and changes in the stacks may be managed using the methods and logical data structures discussed above. In another alternative embodiment, there may be other arrangement for the window stack.

[0126] In yet another embodiment, the systems and methods of this disclosure can be implemented in conjunction with a special purpose computer, a programmed microprocessor or microcontroller and peripheral integrated circuit element(s), an ASIC or other integrated circuit, a digital signal processor, a hard-wired electronic or logic circuit such as discrete element circuit, a programmable logic device or gate array such as PLD, PLA, FPGA, PAL, special purpose computer, any comparable means, or the like. In general, any device(s) or means capable of implementing the methodology illustrated herein can be used to implement the various aspects of this disclosure. Exemplary hardware that can be used for the disclosed embodiments, configurations and aspects includes computers, handheld devices, telephones (e.g., cellular, Internet enabled, digital, analog, hybrids, and others), and other hardware known in the art. Some of these devices include processors (e.g., a single or multiple microprocessors), memory, nonvolatile storage, input devices, and output devices. Furthermore, alternative software implementations including, but not limited to, distributed processing or component/object distributed processing, parallel processing, or virtual machine processing can also be constructed to implement the methods described herein.

[0127] In yet another embodiment, the disclosed methods may be readily implemented in conjunction with software using object or object-oriented software development environments that provide portable source code that can be used on a variety of computer or workstation platforms. Alternatively, the disclosed system may be implemented partially or fully in hardware using standard logic circuits or VLSI design. Whether software or hardware is used to implement the systems in accordance with this disclosure is dependent on the speed and/or

efficiency requirements of the system, the particular function, and the particular software or hardware systems or microprocessor or microcomputer systems being utilized.

[0128] In yet another embodiment, the disclosed methods may be partially implemented in software that can be stored on a storage medium, executed on programmed general-purpose computer with the cooperation of a controller and memory, a special purpose computer, a microprocessor, or the like. In these instances, the systems and methods of this disclosure can be implemented as program embedded on personal computer such as an applet, JAVA® or CGI script, as a resource residing on a server or computer workstation, as a routine embedded in a dedicated measurement system, system component, or the like. The system can also be implemented by physically incorporating the system and/or method into a software and/or hardware system.

[0129] Although the present disclosure describes components and functions implemented in the aspects, embodiments, and/or configurations with reference to particular standards and protocols, the aspects, embodiments, and/or configurations are not limited to such standards and protocols. Other similar standards and protocols not mentioned herein are in existence and are considered to be included in the present disclosure. Moreover, the standards and protocols mentioned herein and other similar standards and protocols not mentioned herein are periodically superseded by faster or more effective equivalents having essentially the same functions. Such replacement standards and protocols having the same functions are considered equivalents included in the present disclosure.

[0130] The present disclosure, in various aspects, embodiments, and/or configurations, includes components, methods, processes, systems and/or apparatus substantially as depicted and described herein, including various aspects, embodiments, configurations embodiments, subcombinations, and/or subsets thereof. Those of skill in the art will understand how to make and use the disclosed aspects, embodiments, and/or configurations afrer understanding the present disclosure. The present disclosure, in various aspects, embodiments, and/or configurations, includes providing devices and processes in the absence of items not depicted and/or described herein or in various aspects, embodiments, and/or configurations hereof, including in the absence of such items as may have been used in previous devices or processes, e.g., for improving performance, achieving ease and\or reducing cost of implementation.

[0131] The foregoing discussion has been presented for purposes of illustration and description. The foregoing is not intended to limit the disclosure to the form or forms disclosed herein. In the foregoing Detailed Description for example, various features of the disclosure are grouped together in one or more aspects, embodiments, and/or configurations for the purpose of streamlining the disclosure. The features of the aspects, embodiments, and/or configurations of the disclosure may be combined in alternate aspects, embodiments, and/or configurations other than those discussed above. This method of disclosure is not to be interpreted as reflecting an intention that the claims require more features than are expressly recited in each claim. Rather, as the following claims reflect, inventive aspects lie in less than all features of a single foregoing disclosed aspect, embodiment, and/or configuration. Thus, the following claims are hereby incorporated into this Detailed Description, with each claim standing on its own as a separate preferred embodiment of the disclosure.

[0132] Moreover, though the description has included description of one or more aspects, embodiments, and/or configurations and certain variations and modifications, other variations, combinations, and modifications are within the scope of the disclosure, e.g., as may be within the skill and knowledge of those in the art, afrer understanding the present disclosure. It is intended to obtain rights which include alternative aspects, embodiments, and/or configurations to the extent permitted, including alternate, interchangeable and/or equivalent structures, functions, ranges or steps to those claimed, whether or not such alternate, interchangeable and/or equivalent structures, functions, ranges or steps are disclosed herein, and without intending to publicly dedicate any patentable subject matter.

What is claimed is:

1. A computer readable medium, having stored thereon, computer-executable instructions executable by a processor, the computer-executable instructions causing the processor to execute a method for creating a window stack for a multi-screen device, the computer-executable instructions comprising:

instructions to receive a gesture to move a window associated with the device;

instructions to determine a display associated with the moved window;

instructions to determine a new position in the window stack for the moved window; and

after determining the position of and the display associated with the moved window, instructions to change a logic data structure associated with the moved window to describe the new position of the moved window in the window stack.

2. The computer readable medium as defined in claim 1, wherein the window stack is a logical arrangement of active and inactive windows for the multi-screen device.

3. The computer readable medium as defined in claim 2, wherein the moved window is displayed on at least one of two displays of the multi-screen device.

4. The computer readable medium as defined in claim 1, wherein the new position in the window stack can be associated with a different display of the at least two displays.

5. The computer readable medium as defined in claim 1, wherein the new position in the window stack can be a position among all windows without association to one of the displays of the multi-screen device.

6. The computer readable medium as defined in claim 1, wherein the logical data structure comprises one or more of:

a window identifier adapted to identify the moved window in relation to other windows in the window stack;

a window stack position identifier adapted to identify the new position in the window stack for the moved window; and

a display identifier adapted to identify which one of two displays of the multi-screen device the moved window is associated.

7. The computer readable medium as defined in claim 6, wherein a first display of the multi-screen device is associated with a first stack and a second display of the multi-screen device is associated with a second stack.

8. The computer readable medium as defined in claim 7, wherein the logic data structure further comprises a window stack identifier adapted to identify which one or more of the first stack and second stack the moved window is associated.

9. The computer readable medium as defined in claim 8, wherein the moved window is displayed across both the first and second display.

10. The computer readable medium as defined in claim 9, wherein the moved window has an identifier for a composite display associated with all the displays of the multi-screen device.

11. A device comprising:

at least two displays;

a memory;

a processor in communication with the memory and each of the at least two displays, the processor operable to:

receive a gesture on the device, wherein the gesture moves a window;

determine a new position in the window stack for the moved window or the one or more altered windows;

determine one of the two displays of the device associated with the moved window or the one or more altered windows; and

after determining the position and associated display of the moved window or the one or more altered windows, change the one or more logic data structures associated with the moved window or the one or more altered windows to describe the new position of the moved window or the one or more altered windows in the window stack, wherein the window stack is an arrangement of the moved window and at least one other window for at least one of the at least two displays.

12. The computer readable medium as defined in claim 11, wherein the moved window is displayed on at least one of the at least two displays of the device.

13. The computer readable medium as defined in claim 11, wherein the position in the window stack can be associated with at least one of the at least two displays or among all windows without association to one of the at least two displays of the device.

14. The computer readable medium as defined in claim 11, wherein the logic data structure comprises one or more of:

a window identifier adapted to identify the moved window in relation to other windows in the window stack;

a window stack position identifier adapted to identify the position in the window stack for the moved window; and

a display identifier adapted to identify which one of the at least two displays of the multi-screen device the moved window is associated.

15. The computer readable medium as defined in claim 14, wherein a first display of the multi-screen device is associated with a first stack and a second display of the multi-screen device is associated with a second stack.

16. A method for changing a window stack in a multi-screen device, the method comprising:

receiving a gesture associated with a window in a first display of the multi-screen device;

interpreting the gesture as moving the window;

determining if the gesture affects one or more other windows;

determining a new position in the window stack for the moved window;

determining a display associated with the moved window;

changing a logic data structure associated with the moved window to describe the new position of the moved window in the window stack

if the gesture affects one or more other windows:

determining a second new position for at least one of the one or more other windows; and

changing a second logic data structure associated with at least one of the one or more other windows to describe the second new position of the at least one of the one or more other windows in the window stack.

17. The method defined in claim 16, wherein the logic data structure comprises one or more of:

a window identifier that identifies the moved window in relation to other windows in the window stack;

a window stack position identifier adapted to identify the new position in the window stack for the moved window; and

a display identifier adapted to identify at least one of the first display or a second display of the multi-screen device is associated with the moved window.

18. The method defined in claim 16, wherein the window stack includes at least one inactive window.

19. The method defined in claim 16, wherein the moved window is also associated with a second display of the multi-screen device.

20. The method defined in claim 19, wherein the logic data structure also comprises a second display identifier adapted to identify that the second display of the multi-screen device is associated with the moved window.

ABSTRACT

Systems and methods are provides for changing a window stack for a multi-screen device. The window stack can change based on the movement of a window. The system can receive a gesture indicating a change in the position of a window in the device. Upon receiving the gesture, the system determines a new position in the window stack for the moved window. Then, the system can determine a display associated with the moved window and change the logic data structure associated with the moved window to describe the new position of the moved window in the window stack.

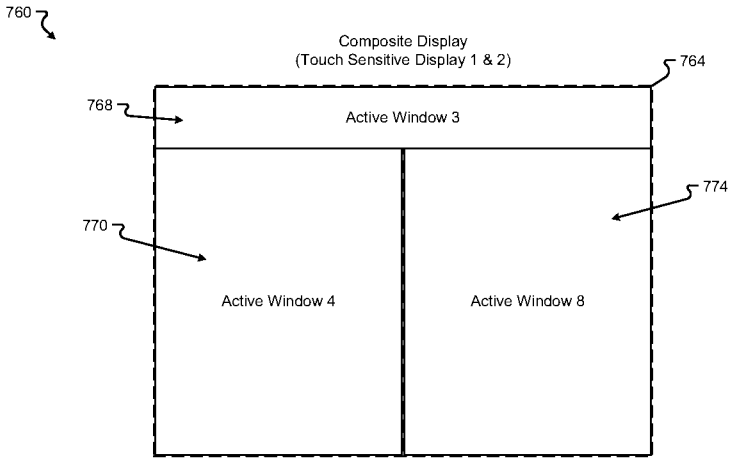


FIG. 7C

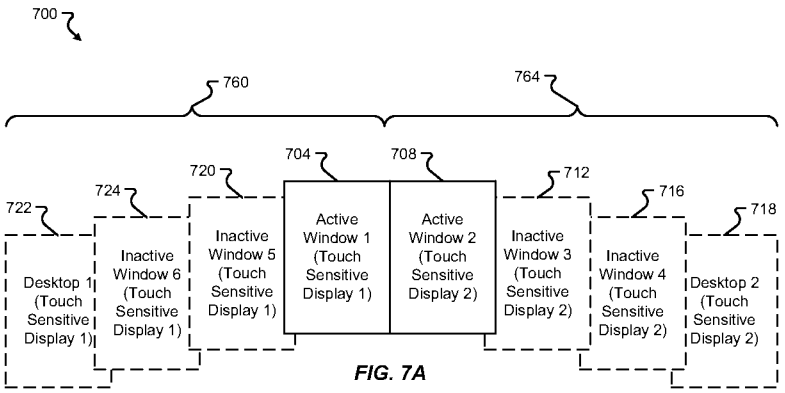


FIG. 7A

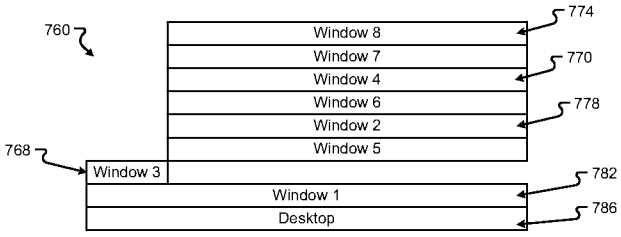


FIG. 7D

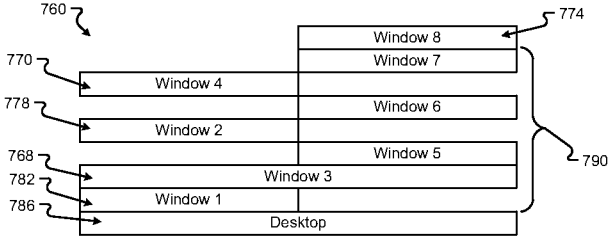


FIG. 7E

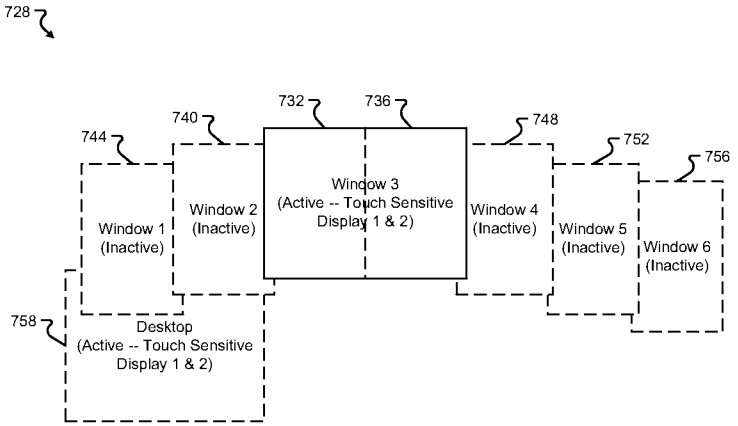


FIG. 7B

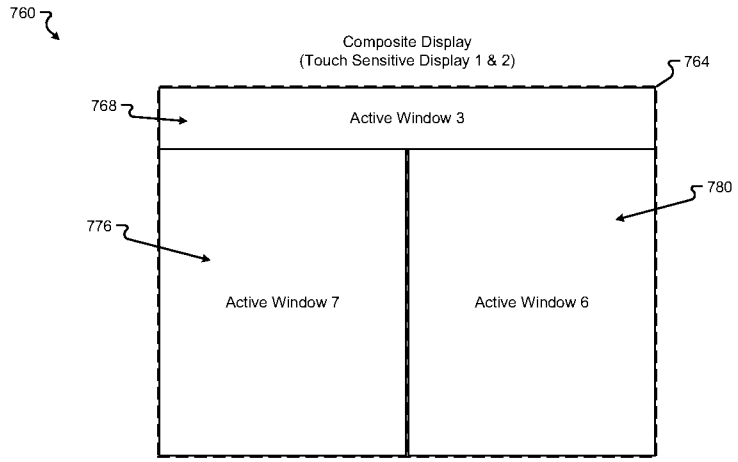


FIG. 7I

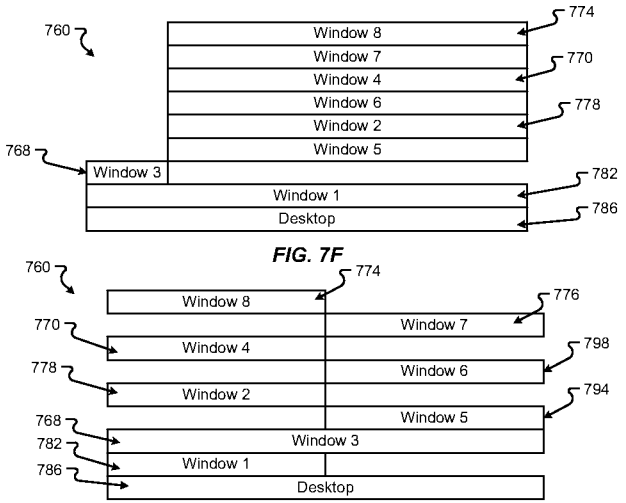


FIG. 7F

FIG. 7G

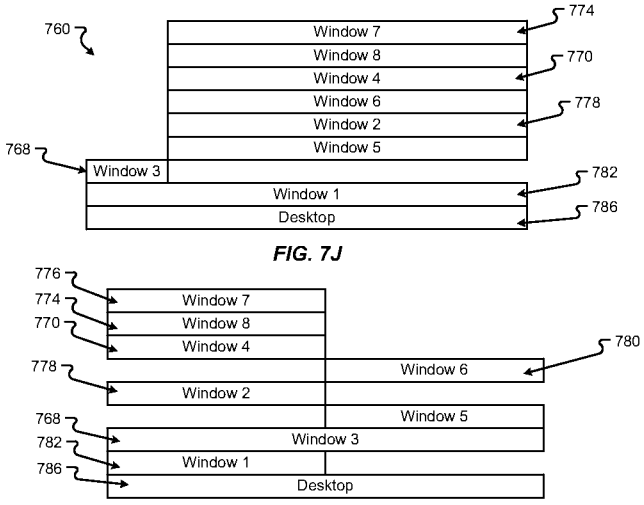


FIG. 7J

FIG. 7K

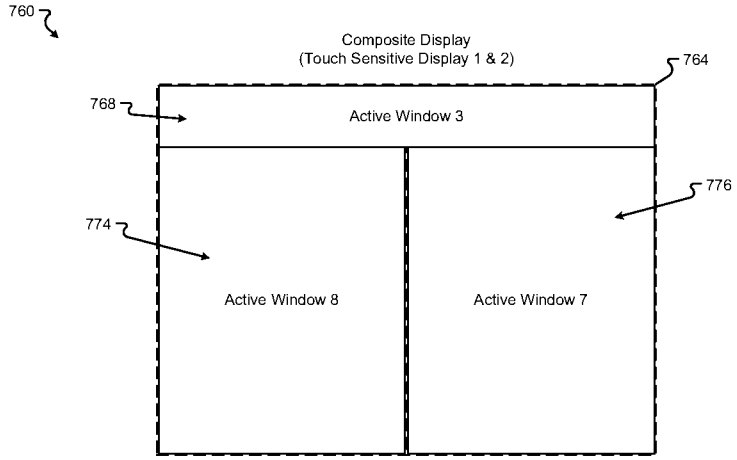


FIG. 7H

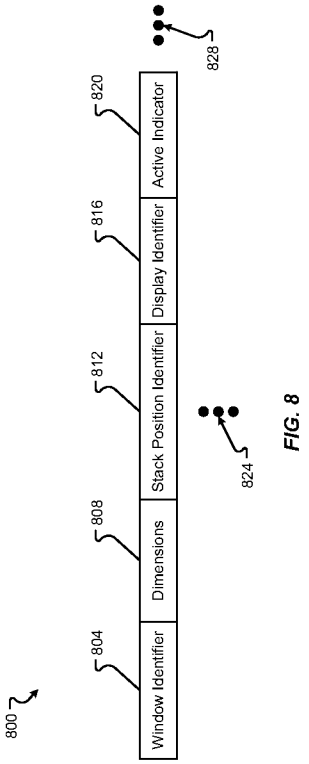

800
804
Window Identifier
808
Dimensions
812
Stack Position Identifier
816
Display Identifier
820
Active Indicator
824
828

FIG. 8

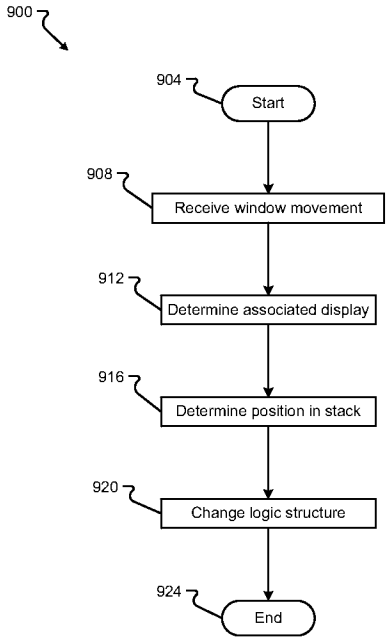

900
904
Start
908
Receive window movement
912
Determine associated display
916
Determine position in stack
920
Change logic structure
924
End

FIG. 9

[0100] The device 100 manages desktops and/or windows with at least one window stack 700, as shown in Figs. 7A and 7B. A window stack 700 is a logical arrangement of active and/or inactive windows or display objects for a multi-screen device. For example, the window stack 700 may be logically similar to a deck of cards or a stack of bricks, where one or more windows or display objects (*e.g.*, desktops) are arranged in an order, as shown in Figs. 7A and 7B. An active window is a window that is currently being displayed on at least one of the touch sensitive displays 110, 114. For example, window 2 712 is an active windows and displayed on only one of the touch sensitive displays 114. In the embodiment shown in Fig. 7A, the device 100 is in a closed state 304 displaying window 2 712 in configuration 608. An inactive window is a window that was opened and displayed but is now "behind" an active window and not being displayed. In embodiments, an inactive window may be for an application that is suspended, and thus, the window is not displaying active content. For example, window 1 708 and window 3 716 are inactive windows. The window stack 700 may be subdivided into portions. For example, as shown in Fig. 7B, window stack 700 can include a first portion 704 and a second portion 724.

[0101] A window stack 700 may have various arrangements or organizational structures. In the embodiment shown in Fig. 7A, the device 100 includes a first stack 704 associated with a first touch sensitive display 114, while the device 100 is in the closed state 304 in the second portrait configuration 608. Thus, each touch sensitive display 114 can have an associated window stack 704. In other embodiments, there is a single window stack that encompasses all of a composite display. The composite display is a logical structure that defines the entire display space including the two touch sensitive displays 110, 114. The device 100 can have a single stack for the composite display where the windows or display objects are sized to occupy a portion or all of the composite display. Thus, the stack portion 704 can represent a portion of a larger composite window stack, part of which is not shown because the device 100 is in a closed state 304.

[0102] In Fig. 7B, the two portions of the window stack (or two window stacks) 704, 724 may have a different number of windows or display objects arranged in the respective stack 704, 724. Further, the two window stack portions 704, 724 can also be identified differently and managed separately. As shown in Fig. 7A, the first window stack portion 704 can be arranged in order ftom a first window 712 to a next window 716 and finally to a desktop 720, which, in

embodiments, is at the "bottom" of the window stack 704. In embodiments, the desktop 720 is not always at the "bottom" as application windows can be arranged in the window stack below the desktop 720, and the desktop 720 can be brought to the "top" of a stack over other windows during a desktop reveal or other orientation change. In embodiments, window 1 708 is also in the stack, but positioned to the left of the active window 712. Thus, window 1 708 may not be view but placed in the stack 700 in a logical position where the user may bring the window 708 back onto the display 114 by a gesture.

[0103] As shown in Fig. 7B, the device 100 transitions to an open state 320. The device 100 transitions to a display configuration as shown in Fig. 6C. As such, the touch sensitive display 110 has no windows associated with the touch sensitive display 110 before the device is opened. However, the logical position of window 1 708 causes the device 100 to associated window 1 708 with the stack portion 724 associated with the touch sensitive display 110. Thus, the second stack 724 can includes on the window 1 708, which, in embodiments, was inactive but becomes an active display when the device is opened. A logical data structure for managing the window stack having two portions 704, 724 may be as described in conjunction with Fig. 8.

[0104] The arrangement of the window stack 704/724 after the device is opened is shown in Figs. 7C through 7E. The window stack 704/724 is shown in three "elevation" views. In Fig. 7C, the top of the window stack 704/724 is shown. Two adjacent sides of the window stack 704/724 are shown in Fig. 7D and 7E. In this embodiment, the window stack 704/724 resembles a stack of bricks. The windows are stacked on each other. Looking ftom the top of the window stack 704/724 in Fig, 7C, only the top most windows in the window stack 704/724 are seen in different portions of a composite display 728. Window 1 708 and/or window 2 712 can occupy part or all of the composite display 728. Fig. C shows a composite display 728 after the device 100 is opened. The composite display 728 encompasses or includes the display area of all the touch sensitive displays 110,114. The size of the composite display 728 can change based on the orientation of the device 100. For example, the size of the composite display 728 of the device 100 when in the closed state may only include the area for one of the touch sensitive displays 110 or 114, as shown in Fig. 7A. The composite display 728 expands to include both touch sensitive displays 110,114 when the device 100 is opened, as shown in Fig. 7B and 7C. Some windows or display objects are associated with the composite display 728 and also change dimensions when the composite display 728 changes dimensions. In embodiments, a window in

a part of the window stack can be associated with the newly exposed display 110 after the device 100 is opened. For example, due to the position of window 1 708 in the window stack 704/724 (to the "left" of the active window, window 1 708 can be associated with the first touch sensitive display 110 after the device is opened.

[0105] In the embodiment shown, the desktop 720 is the lowest display object, window, or "brick" in the window stack 704/724. Thereupon, window 1 708, window 2 712, and window 3 716 are layered. Window 1 708, window 2 712, and window 3 716 only occupy a portion of the composite display 728. Only the top window or display object in any portion 704/724 of the composite display 728 is actually rendered and displayed. Thus, as shown in the top view in Fig. 7C, window 1 708 and window 2 712 are displayed by being at the top of the different portions of the window stack 704/724. A window can be dimensioned to occupy only a portion of the composite display 728 to "reveal" windows lower in the window stack 704/724. For example, the window 1 708 is lower in the stack that window 2 712, but is still displayed. This arrangement of windows and the desktop occurs when the device is opened when displaying the top window in the window stack 704/724.

[0106] Where and how window 1 708 is positioned within the stack can be a function of the orientation of the device 100, the context of what programs, functions, software, etc. are being executed on the device 100, how the stack is positioned when the device 100 is opened, etc. A logical data structure associated with window 1 708 or the other windows may change when the device 100 opens, and the logical data structures can determine how to display the windows and the desktop. When user interface or other events or tasks change the arrangement of the stack, the logical structures of the windows or desktop(s) can be changed to reflect the change in arrangement. In embodiments, the

[0107] Another embodiment of a window stack arrangement 724, which has changed because the device 100 has been opened from a closed configuration shown in Fig. 7F, is shown in Figs. 7G through 7J. In the embodiment shown in Fig. 7F, the device 100 is in a closed state 304 displaying window 2 708 in configuration 608. Window 1 708 and window 3 716 are inactive windows. In the embodiment shown in Fig. 7F, the device 100 shows a second portion 724 of the stack 700 associated with a second touch sensitive display 110, while the device 100 is in the closed state 304 in first portrait configuration 604. Thus, the touch sensitive display 110 can have an associated window stack 724. In other embodiments, there is a single window stack that

encompasses all of a composite display. The device 100 can have a single stack for the composite display where the windows or display objects are sized to occupy a portion or all of the composite display. Thus, the stack 724 can represent a portion of a larger composite display stack, part of which is not shown because the device 100 is in a closed state 304.

[0108] In Fig. 7G, the two portions of the window stack (or two window stacks) 704, 724 may have a different number of windows or display objects arranged in the respective stack portions 704, 724. Further, the two window stack portions 704, 724 can also be identified differently and managed separately. As shown in Fig. 7F, the second window stack 724 can be arranged in order ftom a first window 708 to a next window 712 to a last window 716 and finally to a desktop 720, which, in embodiments, is at the "bottom" of the window stack 704/724. In embodiments, the desktop 720 is not always at the "bottom" as application windows can be arranged in the window stack below the desktop 720, and the desktop 720 can be brought to the "top" of a stack over other windows, such as during an orientation change.

[0109] As shown in Fig. 7G, the device 100 transitions to an open state 320. The device 100 transitions to a display configuration as shown in Fig. 6C. As such, the touch sensitive display 114 has no windows associated with the touch sensitive display 114. However, window 2 712 was not the last window in the window stack 700. Rather, window 1 708 was to the "right" of window 2 712, the active window. Therefore, window 1 708 is displayed for touch sensitive display 114. Thus, the second stack 704 can include the window 1 708. A logical data structure for managing the two window stacks or single window stack, having two portions 704, 724, may be as described in conjunction with Fig. 8.

[0110] The arrangement of the window stack 704/724 after the device is opened is shown in Figs. 7H through 7J. The window stack 704/724 is shown in three "elevation" views. In Fig. 7H, the top of the window stack 704/724 is shown. Two adjacent "sides" of the window stack 704/724 are shown in Fig. 7I and 7J. In this embodiment, the window stack 704/724 resembles a stack of bricks. The windows are stacked on each other. Looking ftom the top of the window stack 704/724 in Fig, 7H, only the top most windows/display objects in the window stack 704/724 are seen in different portions of a composite display 728. A desktop 720 or a window can occupy part or all of the composite display 728.

[0111] In the embodiment shown, the desktop 720 is the lowest display object, window, or "brick" in the window stack 704/724. Thereupon, window 1 708, window 2 712, and window 3

716 are layered. Window 2 712 and window 3 716 only occupy a portion of the composite display 728. Thus, another part of the stack 724 includes only window 1 708. Only the top window or display object in any portion of the composite display 728 is actually rendered and displayed. Thus, as shown in the top view in Fig. 7H, window 1 708 and window 2 712 are displayed by being at the top of the different portions of the window stack 704/724. A window can be dimensioned to occupy only a portion of the composite display 728 to "reveal" windows lower in the window stack 704/724. For example, the window 1 708 is lower in the stack than window 2 712 but is still displayed. This arrangement of windows and the desktop occurs when the device is opened when displaying a top window in the window stack 704/724.

[0112] Where and how window 1 708 is positioned within the stack 704/724 can be a function of the orientation of the device 100, the context of what programs, functions, software, etc. are being executed on the device 100, how the stack is positioned when the device 100 is opened, etc. A logical data structure associated with the window 1 708 may change when the device 100 opens, and the logical data structures can determine how to display the windows and the desktop. When user interface or other events or tasks change the arrangement of the stack, the logical structures of the windows or desktop(s) can be changed to reflect the change in arrangement.

[0113] A logical data structure 800 for managing the arrangement of windows or desktops in a window stack is shown in Fig. 8. The logical data structure 800 can be any data structure used to store data whether an object, record, file, etc. The logical data structure 800 can be stored in any type of database or data storage system, regardless of protocol or standard. In embodiments, the logical data structure 800 includes one or more portions, fields, attributes, etc. that store data in a logical arrangement that allows for easy storage and retrieval of the information. Hereinafrer, these one or more portions, fields, attributes, etc. shall be described simply as fields. The fields can store data for a window identifier 804, dimensions 808, a stack position identifier 812, a display identifier 816, and/or an active indicator 820. Each window in a window stack can have an associated logical data structure 800. While only a single logical data structure 800 is shown in Fig. 8, there may be more or fewer logical data structures 800 used with a window stack (based on the number of windows or desktops in the stack), as represented by ellipses 824. Further, there may be more or fewer fields than those shown in Fig. 8, as represented by ellipses 828.

[0114] A window identifier 804 can include any identifier (ID) that uniquely identifies the associated window or display object in relation to other windows or display objects in the window stack. The window identifier 804 can be a globally unique identifier (GUID), a numeric ID, an alphanumeric ID, or other type of identifier. In embodiments, the window identifier 804 can be one, two, or any number of digits based on the number of windows or display objects that can be opened. In alternative embodiments, the size of the window identifier 804 may change based on the number of windows or display objects opened. While the window or display object is open, the window identifier 804 may be static and remain unchanged.

[0115] Dimensions 808 can include dimensions for a window or display object in the composite display 728. For example, the dimensions 808 can include coordinates for two or more corners of the window or display object or may include one coordinate and dimensions for the width and height of the window or display object. These dimensions 808 can delineate what portion of the composite display 728 the window or display object may occupy, which may the entire composite display 728 or only part of composite display 728. For example, window 1 708 may have dimensions 880 that indicate that the window 1 708 will occupy only part of the display area for composite display 728, as shown in Figs. 7C and 7H. As windows or display objects are moved or inserted in the window stack, the dimensions 808 may change.

[0116] A stack position identifier 812 can be any identifier that can identify the position in the stack for the window or display object or may be inferred from the window's control record within a data structure, such as a list or a stack. The stack position identifier 812 can be a GUID, a numeric ID, an alphanumeric ID, or other type of identifier. Each window or display object can include a stack position identifier 812. For example, as shown in Fig. 7A, window 1 708 in stack 1 704 can have a stack position identifier 812 of 2 identifying that window 1 708 is not the first window in the stack 704 and is an inactive window. Similarly, window 2 712 can have a stack position identifier 812 of 1 representing that window 2 712 is the first window in the stack 704 and active. Thus, depending on the type of stack, the stack position identifier 812 can represent a window's or display object's location in the stack.

[0117] A display identifier 816 can identify that the window or display object is associated with a particular display, such as the first display 110, the second display 114, or the composite display 728 composed of both displays. While this display identifier 816 may not be needed for a multi-stack system, as shown in Fig. 7A, the display identifier 816 can indicate whether a

window or display object in the stack 704/724 of Fig. 7B is displayed on a particular display. Thus, the window 1 708 may have a display identifier 816 for the second portion 724 of the window stack 700 as shown in Fig. 7C. The first portion may have a display identifier 816 for the first display 110 while the second portion may have a display identifier 816 for the second display 114. However, in alternative embodiments, window 1 708 may have a single display identifier 816 identifying the composite display 728.

[0118] Similar to the display identifier 816, an active indicator 820 may not be needed with the single stack system of Fig. 7A, as the window or display object in stack position 1 is active and displayed. In alternative embodiments, the active indicator 820 can indicate which window(s) in the stack is being displayed. Thus, window 1 708 may be shown and have an active indicator 820 when the device is opened and the window is displayed on the touch sensitive display 110,114. The active indicator 820 can be a simple flag or bit that represents that the window or display object is active or displayed.

[0119] An embodiment of a method 900 for changing a window stack is shown in Fig. 9. While a general order for the steps of the method 900 is shown in Fig. 9. Generally, the method 900 starts with a start operation 904 and ends with an end operation 924. The method 900 can include more or fewer steps or can arrange the order of the steps differently than those shown in Fig. 9. The method 900 can be executed as a set of computer-executable instructions executed by a computer system and encoded or stored on a computer readable medium. Hereinafter, the method 900 shall be explained with reference to the systems, components, modules, software, data structures, user interfaces, etc. described in conjunction with Figs. 1-8.

[0120] A multi-screen device 100 can receive an orientation change, as described in Figs. 3A-3B, which changes the device 100 from a closed state 304 to an open state 320, in step 908. The orientation change can be detected and signaled by a hardware input from Hall Effect sensors, a timer, etc. The orientation change can be received by the Task Management Module 540 and sent to the Multi-Display Management Module 524. The Multi-Display Management Module 524 may interpret the change to the configuration of the display from a closed portrait display 604, 608 to an open portrait display (as shown in Fig. 6C) or from a landscape closed display 612, 616 to an open landscape configuration (as shown in Fig. 6F), as described in conjunction with Figs. 6A through 6F. In embodiments, the Task Management Module 540 places the user interface interaction in the task stack 552 to be acted upon by the Multi-Display Management

Module 524. Further, the Task Management Module 540 waits for information from the Multi-Display Management Module 524 to send instructions to the Window Management Module 532 to create the window in the window stack 704.

[0121] The Multi-Display Management Module 524, upon receiving instruction from the Task Management Module 540, determines the window stack disposition, in step 912. In embodiments, window 1 708 is positioned to the side of the active window, window 2 712, as shown in Figs. 7B, 7E or 7F, 7J. The device 100 may be displaying a window that is in the "middle" of the stack 704/724. In other words, a window that is not at the top of the stack 704/724 is being displayed and there are windows in a position to fill the newly exposed display in the open state. For example, in Fig. 7A, window 1 708 is to the "left" of window 2 712. As such, the Multi-Display Management Module 524 can present window 1 708 in display 110, as shown in Fig. 7B. As window 1 708 covers the desktop 720, as shown in Figs. 7E and 7J, the Multi-Display Management Module 524 determines to display window 1 708 in the opened display.

[0122] In embodiments, the device state module 574 of the Multi-Display Management Module 524 may determine how the device is oriented or in what state the device is in, e.g., open, closed, portrait, etc. Further, the preferences module 572 and/or requirements module 580 may determine how windows are to be displayed based on the preferences of the windows. The Display Configuration Module 568 may then use the input from the device state module 574, preferences module 572, and/or other framework components 520 to configure the current window stack 704/724, in step 916. Window 1 708 being shown in the new display typically does not affect other windows in the window stack 704/724 because the other windows do not move in the stack 700. However, the dimensions 808 of window 1 708 can change as the window 1 708 is modified to become active in the composite display 728, which was exposed upon the opening of the device 100.

[0123] The visibility algorithm, in embodiments, determines for all portions of the composite display 728, which windows/display objects are at the top of the stack 704/724 to display the stack. For example, the visibility algorithm determines that, afrer the device 100 is opened, window 1 708 is revealed in one portion of the stack 704, as viewed in Figs. 7B or 7F. Further, window 2 712 is displayed in the other portion of the stack 704, as viewed in Figs. 7B or 7F. Upon determining to display window 1 708 and window 2 712, the Display Configuration

Module 568 can change the display dimensions 808, display identifiers 816, and/or stack position identifiers 812 for the windows. The Multi-Display Management Module 524 can then send the dimensions 808, display identifiers 816, and/or stack position identifiers 812 back to the Task Management Module 540.

[0124] In embodiments, the Task Management Module 540 sends the dimensions 808, display identifiers 816, and/or stack position identifiers 812, and/or other information and instructions to render the windows to the Window Management Module 532. The Window Management Module 532 and the Task Management Module 540 can change the logical data structure(s) 800, in step 920. Both the Task Management Module 540 and the Window Management Module 532 may manage copies of the window stack 704/724. These copies of the window stack 704/724 can be synchronized or kept similar through communications between the Window Management Module 532 and the Task Management Module 540. Thus, the Window Management Module 532 and the Task Management Module 540, based on the information determined by the Multi-Display Management Module 524, can change dimensions 808, display identifiers 816, and/or stack position identifiers 812 for the one or more window(s). The logical data structure 800 may then be stored by both the Window Management Module 532 and the Task Management Module 540. Further, the Window Management Module 532 and the Task Management Module 540 may thereinafrer manage the window stack 704/724 and the logical data structure(s) 800.

[0125] The exemplary systems and methods of this disclosure have been described in relation to a window stack associated with a multi-screen device. However, to avoid unnecessarily obscuring the present disclosure, the preceding description omits a number of known structures and devices. This omission is not to be construed as a limitation of the scopes of the claims. Specific details are set forth to provide an understanding of the present disclosure. It should however be appreciated that the present disclosure may be practiced in a variety of ways beyond the specific detail set forth herein.

[0126] Furthermore, while the exemplary aspects, embodiments, and/or configurations illustrated herein show the various components of the system collocated, certain components of the system can be located remotely, at distant portions of a distributed network, such as a LAN and/or the Internet, or within a dedicated system. Thus, it should be appreciated, that the components of the system can be combined in to one or more devices, such as the software modules of Figs. 5A and 5B, or collocated on a particular node of a distributed network, such as

an analog and/or digital telecommunications network, a packet-switch network, or a circuit-switched network. It will be appreciated from the preceding description, and for reasons of computational efficiency, that the components of the system can be arranged at any location within a distributed network of components without affecting the operation of the system. For example, the various components can be located in a switch such as a PBX and media server, gateway, in one or more communications devices, at one or more users' premises, or some combination thereof. Similarly, one or more functional portions of the system could be distributed between a telecommunications device(s) and an associated computing device.

[0127] Furthermore, it should be appreciated that the various links connecting the elements can be wired or wireless links, or any combination thereof, or any other known or later developed element(s) that is capable of supplying and/or communicating data to and from the connected elements. These wired or wireless links can also be secure links and may be capable of communicating encrypted information. Transmission media used as links, for example, can be any suitable carrier for electrical signals, including coaxial cables, copper wire and fiber optics, and may take the form of acoustic or light waves, such as those generated during radio-wave and infra-red data communications.

[0128] Also, while the flowcharts have been discussed and illustrated in relation to a particular sequence of events, it should be appreciated that changes, additions, and omissions to this sequence can occur without materially affecting the operation of the disclosed embodiments, configuration, and aspects.

[0129] A number of variations and modifications of the disclosure can be used. It would be possible to provide for some features of the disclosure without providing others. For example in one alternative embodiment, the window stack 700 resembles a carousel or rolodex rather than a deck of cards. Thus, windows may circulate from one touch sensitive display 110 to another touch sensitive display 114. For example, window may be push to the right and end up at the end of stack behind another window. If the stack continues to move to the right, eventually the window would appear on the second touch sensitive display even though the window was opened in the first touch sensitive display 110. These movements and changes in the stacks may be managed using the methods and logical data structures discussed above. In another alternative embodiment, there may be other arrangement for the window stack.

[0130] In yet another embodiment, the systems and methods of this disclosure can be implemented in conjunction with a special purpose computer, a programmed microprocessor or microcontroller and peripheral integrated circuit element(s), an ASIC or other integrated circuit, a digital signal processor, a hard-wired electronic or logic circuit such as discrete element circuit, a programmable logic device or gate array such as PLD, PLA, FPGA, PAL, special purpose computer, any comparable means, or the like. In general, any device(s) or means capable of implementing the methodology illustrated herein can be used to implement the various aspects of this disclosure. Exemplary hardware that can be used for the disclosed embodiments, configurations and aspects includes computers, handheld devices, telephones (e.g., cellular, Internet enabled, digital, analog, hybrids, and others), and other hardware known in the art. Some of these devices include processors (e.g., a single or multiple microprocessors), memory, nonvolatile storage, input devices, and output devices. Furthermore, alternative software implementations including, but not limited to, distributed processing or component/object distributed processing, parallel processing, or virtual machine processing can also be constructed to implement the methods described herein.

[0131] In yet another embodiment, the disclosed methods may be readily implemented in conjunction with software using object or object-oriented software development environments that provide portable source code that can be used on a variety of computer or workstation platforms. Alternatively, the disclosed system may be implemented partially or fully in hardware using standard logic circuits or VLSI design. Whether software or hardware is used to implement the systems in accordance with this disclosure is dependent on the speed and/or efficiency requirements of the system, the particular function, and the particular software or hardware systems or microprocessor or microcomputer systems being utilized.

[0132] In yet another embodiment, the disclosed methods may be partially implemented in software that can be stored on a storage medium, executed on programmed general-purpose computer with the cooperation of a controller and memory, a special purpose computer, a microprocessor, or the like. In these instances, the systems and methods of this disclosure can be implemented as program embedded on personal computer such as an applet, JAVA® or CGI script, as a resource residing on a server or computer workstation, as a routine embedded in a dedicated measurement system, system component, or the like. The system can also be

implemented by physically incorporating the system and/or method into a software and/or hardware system.

[0133] Although the present disclosure describes components and functions implemented in the aspects, embodiments, and/or configurations with reference to particular standards and protocols, the aspects, embodiments, and/or configurations are not limited to such standards and protocols. Other similar standards and protocols not mentioned herein are in existence and are considered to be included in the present disclosure. Moreover, the standards and protocols mentioned herein and other similar standards and protocols not mentioned herein are periodically superseded by faster or more effective equivalents having essentially the same functions. Such replacement standards and protocols having the same functions are considered equivalents included in the present disclosure.

[0134] The present disclosure, in various aspects, embodiments, and/or configurations, includes components, methods, processes, systems and/or apparatus substantially as depicted and described herein, including various aspects, embodiments, configurations embodiments, subcombinations, and/or subsets thereof. Those of skill in the art will understand how to make and use the disclosed aspects, embodiments, and/or configurations afrer understanding the present disclosure. The present disclosure, in various aspects, embodiments, and/or configurations, includes providing devices and processes in the absence of items not depicted and/or described herein or in various aspects, embodiments, and/or configurations hereof, including in the absence of such items as may have been used in previous devices or processes, e.g., for improving performance, achieving ease and\or reducing cost of implementation.

[0135] The foregoing discussion has been presented for purposes of illustration and description. The foregoing is not intended to limit the disclosure to the form or forms disclosed herein. In the foregoing Detailed Description for example, various features of the disclosure are grouped together in one or more aspects, embodiments, and/or configurations for the purpose of streamlining the disclosure. The features of the aspects, embodiments, and/or configurations of the disclosure may be combined in alternate aspects, embodiments, and/or configurations other than those discussed above. This method of disclosure is not to be interpreted as reflecting an intention that the claims require more features than are expressly recited in each claim. Rather, as the following claims reflect, inventive aspects lie in less than all features of a single foregoing disclosed aspect, embodiment, and/or configuration. Thus, the following claims are hereby

incorporated into this Detailed Description, with each claim standing on its own as a separate preferred embodiment of the disclosure.

[0136] Moreover, though the description has included description of one or more aspects, embodiments, and/or configurations and certain variations and modifications, other variations, combinations, and modifications are within the scope of the disclosure, e.g., as may be within the skill and knowledge of those in the art, after understanding the present disclosure. It is intended to obtain rights which include alternative aspects, embodiments, and/or configurations to the extent permitted, including alternate, interchangeable and/or equivalent structures, functions, ranges or steps to those claimed, whether or not such alternate, interchangeable and/or equivalent structures, functions, ranges or steps are disclosed herein, and without intending to publicly dedicate any patentable subject matter.

What is claimed is:

1. A computer readable medium, having stored thereon, computer-executable instructions executable by a processor, the computer-executable instructions causing the processor to execute a method for managing a window stack for a device, the computer-executable instructions comprising:

instructions to receive an orientation change in the device, wherein the orientation change causes the device to transition from a closed state to an open state, wherein, in the closed state, a first display is active and, in the open state, both the first display and a second display are active;

instructions to determine if a first window should be displayed on the second display; and

afrer determining the first window should be displayed on the second display, instructions to display the first window on the second display and a second window on the first display.

2. The computer readable medium as defined in claim 1, wherein the window stack is a logical arrangement of active and inactive windows for the device.

3. The computer readable medium as defined in claim 2, wherein the window stack provides a representation of windows or display objects to the user.

4. The computer readable medium as defined in claim 3, wherein the first window is inactive in the closed state.

5. The computer readable medium as defined in claim 4, wherein the first window is displayed in a portion of a composite display, wherein the composite display occupies substantially all of the at least the two or more touch sensitive displays, and wherein the first display and second display are portions of the composited display.

6. The computer readable medium as defined in claim 5, further comprising instructions to modify a logical data structure associated with the first window.

7. The computer readable medium as defined in claim 6, wherein the logical data structure comprises one or more of:

a window identifier adapted to identify the first window in relation to other windows in the window stack;

a window stack position identifier adapted to identify the position in the window stack for the first window; and

a display identifier adapted to identify a first touch sensitive display, which is a portion of the composite display of the device, wherein the first window is associated with the first touch sensitive display.

8. The computer readable medium as defined in claim 6, wherein a second display of the device is associated with a second portion of the window stack and a second display of the device is associated with a second portion of the window stack.

9. The computer readable medium as defined in claim 8, wherein the logic data structure further comprises a window stack identifier adapted to identify the first portion of which the first window is associated.

10. A device comprising:

at least two displays;

a memory;

a processor in communication with the memory and each of the at least two displays, the processor operable to:

provide a composite display, wherein the composite display comprises one touch sensitive display in a closed state, wherein a first window is displayed in the composite display in the closed state;

receive an orientation change in the device, wherein the orientation change causes the device to transition from a closed state to an open state, wherein, in the open state, the composite display comprises a first portion associated with the first touch sensitive display and a second portion associated with a second touch sensitive display;

determine the first window is displayed in the first portion of the composite display and a second window is displayed in the second portion of the composite display;

display the first window on the first touch sensitive display; and

display the second window on the second touch sensitive display.

11. The device as defined in claim 10, wherein the second window is altered upon the transition to the open state.

12. The device as defined in claim 11, further comprising altering a logic data structure associated with the second window.

13. The device as defined in claim 12, wherein the second window is inactive during the closed state.

14. The device as defined in claim 13, wherein the logic data structure comprises one or more of:

a window identifier adapted to identify the second window in relation to other windows in the window stack;

a window stack position identifier adapted to identify the top position in the window stack for the second window in at least the second portion of the composite display; and

at least one display identifiers adapted to identify the second touch sensitive display of the device with which the second window is associated.

15. The device as defined in claim 14, wherein the second window is displayed because no window covers the second window in at least the second portion of the composite display.

16. The device as defined in claim 15, wherein the second window covers at least a portion of the desktop.

17. A method for presenting a display for a device, the method comprising:

providing a composite display that spans at least a portion of a first touch sensitive display when the device is in a closed state;

displaying a first window at a top of a window stack when the device is in the closed state;

receiving an orientation change for the device, wherein the orientation change is a transition from the closed state to an open state;

changing the composite display to span at least a portion of the first touch sensitive display and a second touch sensitive display of the device, wherein a first portion of the composite display is associated with the first touch sensitive display and a second portion of the composite display is associated with the second touch sensitive display;

determining that the first window is associated with the first portion of the composite display;

determining that a second window is associated with the second portion of the composite display;

modifying the second window to be displayed in the second portion of the composite display;

determining that the first window is at the top of the window stack in the first portion of the composite display;

determining the second window is at the top of the window stack in the second portion of the composite display;

displaying the first window on the first touch sensitive display; and

displaying the second window on the second touch sensitive display.

18. The method defined in claim 17, wherein the second window is at a top of the window stack in the second portion of the composite display.

19. The method defined in claim 18, wherein the second window is inactive in the closed state.

20. The method defined in claim 19, wherein the second touch sensitive display is part of a primary screen.

ABSTRACT

Systems and methods are provides for displaying a second window for a multi-screen device in response to opening the device. The window stack can change based on the change in the orientation of the device. The system can receive an orientation change that transitions the device from a closed state to an open state. A previously created, but inactive, window in the stack can be displayed on one of the two or more displays comprising the device when opened. The previously created window become active to be displayed on the second of the displays afrer the device is opened.

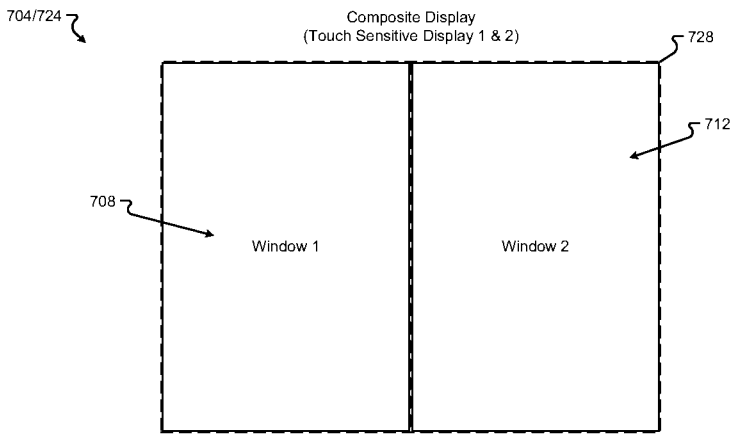


FIG. 7C

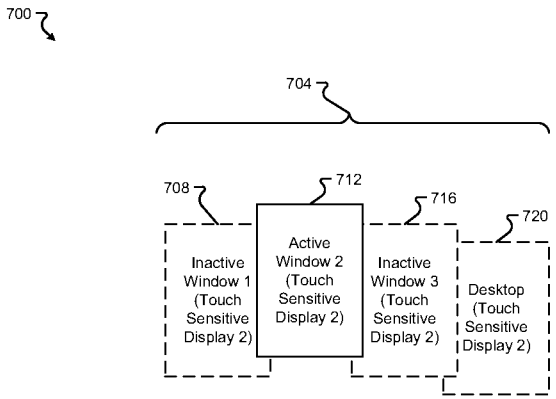


FIG. 7A

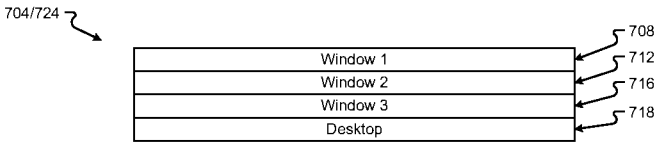


FIG. 7D

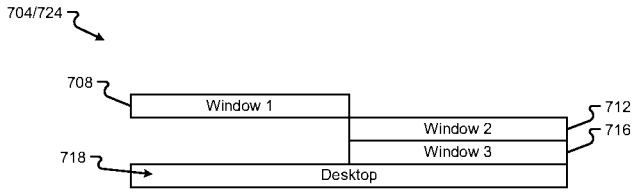


FIG. 7E

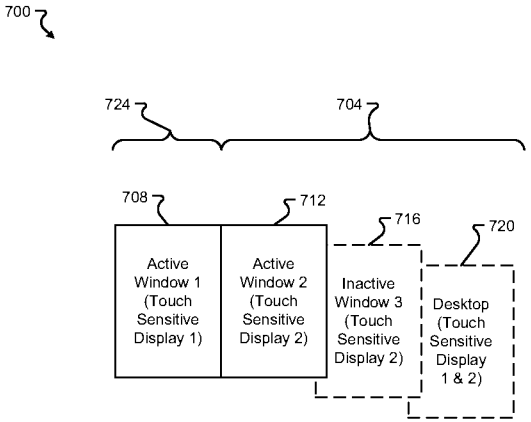
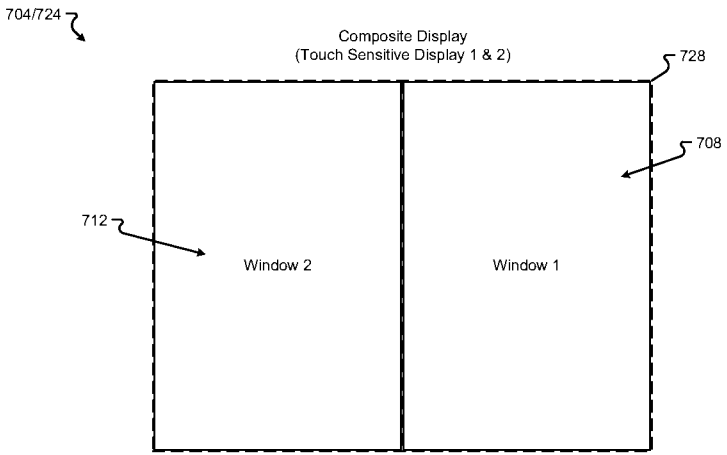


FIG. 7B



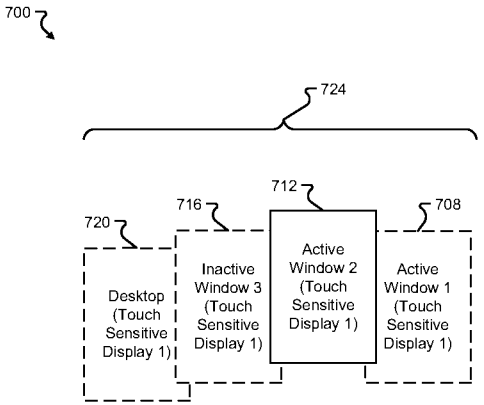


FIG. 7F

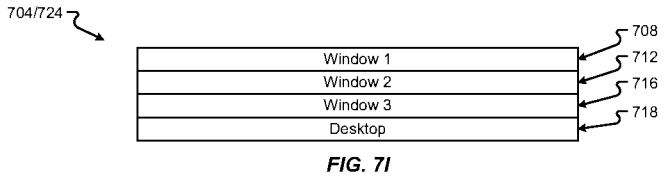


FIG. 7I

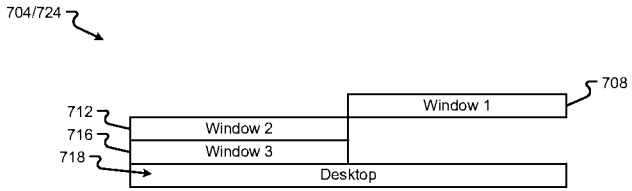


FIG. 7J

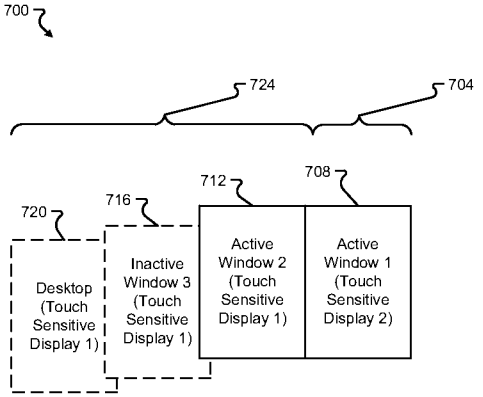


FIG. 7G

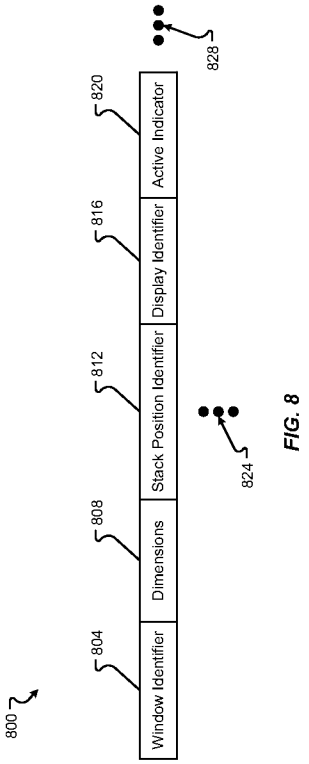

800
804
Window Identifier
808
Dimensions
812
Stack Position Identifier
816
Display Identifier
820
Active Indicator
824
828

FIG. 8

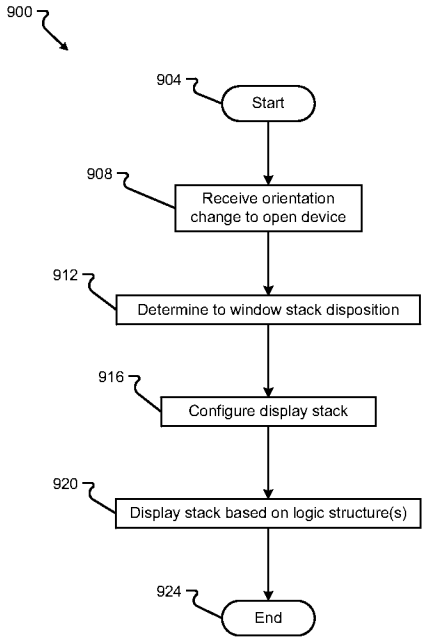

900
904
Start
908
Receive orientation change to open device
912
Determine to window stack disposition
916
Configure display stack
920
Display stack based on logic structure(s)
924
End

FIG. 9

[0100] The device 100 manages desktops and/or windows with at least one window stack 700, 728, as shown in Figs. 7A and 7B. A window stack 700, 728 is a logical arrangement of active and/or inactive windows for a multi-screen device. For example, the window stack 700, 728 may be logically similar to a deck of cards or a stack of bricks, where one or more windows or desktops are arranged in order, as shown in Figs. 7A and 7B. An active window is a window that is currently being displayed on at least one of the touch sensitive displays 110, 114. For example, windows 104 and 108 are active windows and are displayed on touch sensitive displays 110 and 114. An inactive window is a window that was opened and displayed but is now "behind" an active window and not being displayed. In embodiments, an inactive window may be for an application that is suspended, and thus, the window is not displaying active content. For example, windows 712, 716, 720, and 724 are inactive windows.

[0101] A window stack 700, 728 may have various arrangements or organizational structures. In the embodiment shown in Fig. 7A, the device 100 includes a first stack 760 associated with a first touch sensitive display 110 and a second stack associated with a second touch sensitive display 114. Thus, each touch sensitive display 110, 114 can have an associated window stack 760, 764. These two window stacks 760, 764 may have different numbers of windows arranged in the respective stacks 760, 764. Further, the two window stacks 760, 764 can also be identified differently and managed separately. Thus, the first window stack 760 can be arranged in order ftom a first window 704 to a next window 720 to a last window 724 and finally to a desktop 722, which, in embodiments, is at the "bottom" of the window stack 760. In embodiments, the desktop 722 is not always at the "bottom" as application windows can be arranged in the window stack below the desktop 722, and the desktop 722 can be brought to the "top" of a stack over other windows during a desktop reveal. Likewise, the second stack 764 can be arranged from a first window 708 to a next window 712 to a last window 716, and finally to a desktop 718, which, in embodiments, is a single desktop area, with desktop 722, under all the windows in both window stack 760 and window stack 764. A logical data structure for managing the two window stacks 760, 764 may be as described in conjunction with Fig. 8.

[0102] Another arrangement for a window stack 728 is shown in Fig. 7B. In this embodiment, there is a single window stack 728 for both touch sensitive displays 110, 114. Thus, the window stack 728 is arranged ftom a desktop 758 to a first window 744 to a last window 756. A window can be arranged in a position among all windows without an association to a specific touch

sensitive display 110, 114. In this embodiment, a window is in the order of windows. Further, at least one window is identified as being active. For example, a single window may be rendered in two portions 732 and 736 that are displayed on the first touch sensitive screen 110 and the second touch sensitive screen 114. The single window may only occupy a single position in the window stack 728 although it is displayed on both displays 110, 114.

[0103] Yet another arrangement of a window stack 760 is shown in Figs. 7C through 7E. The window stack 760 is shown in three "elevation" views. In Fig. 7C, the top of the window stack 760 is shown. Two sides of the window stack 760 are shown in Figs. 7D and 7E. In this embodiment, the window stack 760 resembles a stack of bricks. The windows are stacked on each other. Looking from the top of the window stack 760 in Fig, 7C, only the top most windows in the window stack 760 are seen in different portions of the composite display 764. The composite display 764 represents a logical model for the entire display area of the device 100, which can include touch sensitive display 110 and touch sensitive display 114. A desktop 786 or a window can occupy part or all of the composite display 764.

[0104] In the embodiment shown, the desktop 786 is the lowest display or "brick" in the window stack 760. Thereupon, window 1 782, window 2 782, window 3 768, and window 4 770 are layered. Window 1 782, window 3 768, window 2 782, and window 4 770 only occupy a portion of the composite display 764. Thus, another part of the stack 760 includes window 8 774 and windows 5 through 7 shown in section 790. Only the top window in any portion of the composite display 764 is actually rendered and displayed. Thus, as shown in the top view in Fig. 7C, window 4 770, window 8 774, and window 3 768 are displayed as being at the top of the display in different portions of the window stack 760. A window can be dimensioned to occupy only a portion of the composite display 760 to "reveal" windows lower in the window stack 760. For example, window 3 768 is lower in the stack than both window 4 770 and window 8 774 but is still displayed.

[0105] When a new window is created because an application is executed or an application execution is changed, the new window is generally positioned at the top of the stack. Herein, the new window can display content across both touch sensitive displays 110/114. In embodiments, the device 100 can display the new window in a Portrait-Max (PMax) configuration 624 or a Landscape-Max (LMax) configuration 628. In these configurations, a single new window is displayed across both touch sensitive displays 110/114. In an alternative embodiment, the new

application is displayed in the Dual-Portrait (PD) output configuration or Dual-Landscape (LD) output configuration but has a parent and child window opened, where the parent window is displayed in a first touch sensitive display 110 and a child window is displayed in a second touch sensitive display 114. This second configuration is shown in Fig. 7F.

[0106] Where and how the window is positioned within the stack can be a function of the orientation of the device 100, the context of what programs, functions, software, etc. are being executed on the device 100, how the stack is positioned when the new window is opened, etc. To create the window in the stack, the state of the stack for the already opened windows may be determined. With this information, a logical data structure associated with the windows can be changed. When user interface or other events or tasks change the arrangement of windows, the window stack(s) can be changed to reflect the change in arrangement.

[0107] An embodiment of a window stack arrangement 760, which has changed because a new window opened (possibly because a new application is executed), is shown in Figs. 7F through 7I. The window stack 760 is shown in a next state (from Figs. 7C through 7E) in four views, Figs. 7F through 7I. The top of the window stack 760 is shown in Fig. 7F, which is a view of what is displayed on the device 100. The two "sides" of the window stack 760 are shown in Figs. 7G and 7H. In this embodiment, windows 9.1 and 9.2 have been opened by a gesture received from the touch sensitive display 110, 114 or from the gesture capture region 120, 124. In embodiments, window 9.1 742 is a parent window and window 9.2 746 is a child window. A child window is a second window opened by an application that can provide further information to a user. For example, when executing an email program, a parent window may display a list of message in an inbox. The child window may display the content of or information about one particular message selected in the inbox.

[0108] In interpreting the gesture to open the window(s) or execute the application, the Task Management Module 540 can request the Multi-Display Management Module 524 to determine where to position new windows 9.1 742 and 9.2 746. The Multi-Display Management Module 524 may then determine the dimensions of the window(s) to position the window(s) at the top of the stack 760 and displayed on both the first touch sensitive display 110 and the second touch sensitive display 110. Thus, looking from the top of the window stack 760 in Figs. 7C and 7F, new window 9.1 742 opens on the left side sided of the composite display 764, while new window 9.2 746 opens on the right side of the stack 760 (shown in Fig. 7H and 7F).

[0109] Based on the opening of new windows 9.1 742 and 9.2 746, the desktop 786 is the lowest window/display object or "brick" in the window stack 760. Thereupon, window 1 782, window 2 782, window 3 768, window 4 770, and new window 9.1 742 are layered on the left side of the composite display or left side of the stack 760. Window 1 782, window 3 768, window 2 782, and window 4 770 only occupy a portion of the composite display 764. Thus, another part the stack 760 includes window 5 794, window 6 798, window 7 776, window 8 774, and new window 9.2 746 on the right side of the composite display 764 or the right side of the stack.

[0110] 760. Only the top window in any portion of the composite display 764 is actually rendered and displayed. Thus, as shown in the top view in Fig. 7F, new windows 9.1 742 and 9.2 746 are displayed. The opening of new windows 9.1 742 and 9.2 746 covered window 8 774 window 4 770, and window 3 768. The new windows 9.1 742 and 9.2 746 are at different levels of the stack 760 and are in different portions of the window stack 760. A window can be dimensioned to occupy only a portion of the composite display 760 to "reveal" windows lower in the window stack 760.

[0111] Fig. 7I provides a logical view, as would be understood by a user, of the window stack 760 after the new windows 9.1 742 and 9.2 746 were opened. Here, new windows 9.1 742 and 9.2 746 appear to occupy both the first touch sensitive display 110 and the second touch sensitive display 114. The other widows would seem to be to the left or right of the new windows 9.1 742 and 9.2 746, as shown in Fig. 7I. Window 3 768 and the desktop 786 would seem to extend across both displays. As such, if a user moved window 9.1 742 to the right, window 4 would appear to move onto the first touch sensitive display 110 ftom the left side of the first touch sensitive display 110. Likewise, if the user closed the child window 9.2 746, window 8 would appear to move from the right and onto the second touch sensitive display 114. Thus, the window stack 760 maintains a "deck of cards" type structure for the user.

[0112] Still another embodiment of a window stack arrangement 760 that is changed because of a single new window 9 750 opens across the entire composited display 764, is shown in Figs. 7J through 7M. The window stack 760 is shown in a next state (ftom Figs. 7C through 7E) in three views, Figs. 7J, 7K, and 7L. The top of the window stack 760 is shown in Fig. 7J, which is a view of what is displayed on the device 100. The two "sides" of the window stack 760 are shown in Figs. 7K and 7L. In this embodiment, window 9 750 is opened by a gesture received

ftom the touch sensitive display 110, 114 or from the gesture capture region 120, 124. In interpreting the gesture, the Task Management Module 540 can request the Multi-Display Management Module 524 to determine where to position window 9 750. The Multi-Display Management Module 524 may then determine the dimensions of window 9 750 to position the window at the top of the stack 760 and displayed across the first touch sensitive display 110 and the second touch sensitive display 114. Thus, looking ftom the side of the window stack 760 in Fig. 7K and 7L, new window 9 750 opens over the right side of the composite display 764 and the left side of the composite display 764. Thus, new window 9 750 covers window 8 774, window 4 770, and window 3 768.

[0113] Based on the opening if new window 9 750, desktop 786 is the lowest window/display object or "brick" in the window stack 760. Thereupon, window 1 782, window 2 782, window 3 768, window 4 770, and a first portion of new window 9 750 are layered on the right side of the stack 760. Window 1 782, window 3 768, window 2 782, and window 4 770 only occupy a portion of the composite display 764. Thus, another part of the stack 760 includes window 5 794, window 6 798, window 7 776, window 8 774, and a second portion of new window 9.2 746 on the right side of the right side of the stack 760. Thus, new window 9 750 occupies both portions of the composite display 764. Only the top window in any portion of the composite display 764 is actually rendered and displayed. Thus, as shown in the top view in Fig. 7J, new window 9 750 is displayed but on both the right and left side of the composite display 764. The opening of new window 9 750 covered window 4 770, window 8 774, and window 3 768.

[0114] Fig. 7M provides a logical view, as would be understood by a user, of the window stack 760 after the new window 9 750 was opened. Here, new window 9 750 appears to occupy both the first touch sensitive display 110 and the second touch sensitive display 114. The other widows would seem to be to the left or right of the new window 9 750, as shown in Fig. 7M. Window 3 768 and the desktop 786 would seem to extend across both displays. As such, if a user moved new window 9 750 to the right, window 4 would appear to move onto the first touch sensitive display 110 ftom the left side of the first touch sensitive display 110. Likewise, if the user closed new window 9 750, window 8 would appear to move from the right and onto the second touch sensitive display 114 while window 4 770 would appear to move onto the first touch sensitive display 110. Thus, the window stack 760 maintains a "deck of cards" type

structure for the user. A logical data structure to manage the window stack can be as described in conjunction with Fig. 8.

[0115] A logical data structure 800 for managing the arrangement of windows or desktops in a window stack is shown in Fig. 8. The logical data structure 800 can be any data structure used to store data whether an object, record, file, etc. The logical data structure 800 can be stored in any type of database or data storage system, regardless of protocol or standard. In embodiments, the logical data structure 800 includes one or more portions, fields, attributes, etc. that store data in a logical arrangement that allows for easy storage and retrieval of the information. Hereinafter, these one or more portions, fields, attributes, etc. shall be described simply as fields. The fields can store data for a window identifier 804, dimensions 808, a stack position identifier 812, a display identifier 816, and/or an active indicator 820. Each window in a window stack can have an associated logical data structure 800. While only a single logical data structure 800 is shown in Fig. 8, there may be more or fewer logical data structures 800 used with a window stack (based on the number of windows or desktops in the stack), as represented by ellipses 824. Further, there may be more or fewer fields than those shown in Fig. 8, as represented by ellipses 828.

[0116] A window identifier 804 can include any identifier (ID) that uniquely identifies the associated window in relation to other windows in the window stack. The window identifier 804 can be a globally unique identifier (GUID), a numeric ID, an alphanumeric ID, or other type of identifier. In embodiments, the window identifier 804 can be one, two, or any number of digits based on the number of windows that can be opened. In alternative embodiments, the size of the window identifier 804 may change based on the number of windows opened. While the window is open, the window identifier 804 may be static and remain unchanged.

[0117] Dimensions 808 can include dimensions for a window in the composite display 760. For example, the dimensions 808 can include coordinates for two or more corners of the window or may include one coordinate and dimensions for the width and height of the window. These dimensions 808 can delineate what portion of the composite display 760 the window may occupy, which may the entire composite display 760 or only part of composite display 760. For example, window 4 770 may have dimensions 880 that indicate that the window 770 will occupy only part of the display area for composite display 760, as shown in Figs. 7c through 7E. As windows are moved or inserted in the window stack, the dimensions 808 may change.

[0118] A stack position identifier 812 can be any identifier that can identify the position in the stack for the window or may be inferred from the window's control record within a data structure, such as a list or a stack. The stack position identifier 812 can be a GUID, a numeric ID, an alphanumeric ID, or other type of identifier. Each window or desktop can include a stack position identifier 812. For example, as shown in Fig. 7A, window 1 704 in stack 1 760 can have a stack position identifier 812 of 1 identifying that window 704 is the first window in the stack 760 and the active window. Similarly, window 6 724 can have a stack position identifier 812 of 3 representing that window 724 is the third window in the stack 760. Window 2 708 can also have a stack position identifier 812 of 1 representing that window 708 is the first window in the second stack 764. As shown in Fig. 7B, window 1 744 can have a stack position identifier 812 of 1, window 3, rendered in portions 732 and 736, can have a stack position identifier 812 of 3, and window 6 756 can have a stack position identifier 812 of 6. Thus, depending on the type of stack, the stack position identifier 812 can represent a window's location in the stack.

[0119] A display identifier 816 can identify that the window or desktop is associated with a particular display, such as the first display 110 or the second display 114, or the composite display 760 composed of both displays. While this display identifier 816 may not be needed for a multi-stack system, as shown in Fig. 7A, the display identifier 816 can indicate whether a window in the serial stack of Fig. 7B is displayed on a particular display. Thus, window 3 may have two portions 732 and 736 in Fig. 7B. The first portion 732 may have a display identifier 816 for the first display while the second portion 736 may have a display identifier 816 for the second display 114. However, in alternative embodiments, the window may have two display identifier 816 that represent that the window is displayed on both of the displays 110, 114, or a display identifier 816 identifying the composite display. In another alternate embodiment, the window may have a single display identifier 816 to represent that the window is displayed on both of the displays 110, 114.

[0120] Similar to the display identifier 816, an active indicator 820 may not be needed with the dual stack system of Fig. 7A, as the window in stack position 1 is active and displayed. In the system of Fig. 7B, the active indicator 820 can indicate which window(s) in the stack is being displayed. Thus, window 3 may have two portions 732 and 736 in Fig. 7. The first portion 732 may have an active indicator 820 while the second portion 736 may also have an active indicator 820. However, in alternative embodiments, window 3 may have a single active indicator 820.

The active indicator 820 can be a simple flag or bit that represents that the window is active or displayed.

[0121] An embodiment of a method 900 for changing a window stack is shown in Fig. 9. While a general order for the steps of the method 900 is shown in Fig. 9. Generally, the method 900 starts with a start operation 904 and ends with an end operation 928. The method 900 can include more or fewer steps or can arrange the order of the steps differently than those shown in Fig. 9. The method 900 can be executed as a set of computer-executable instructions executed by a computer system and encoded or stored on a computer readable medium. Hereinafter, the method 900 shall be explained with reference to the systems, components, modules, software, data structures, user interfaces, etc. described in conjunction with Figs. 1-8.

[0122] A multi-screen device 100 can receive a gesture, as described in Figs. 4A-4H, directed to executing an application that includes opening a window, in step 908. The gesture can be a hardware input ftom a touch sensitive display 110, 114, ftom a gesture capture region 120, 124, or some other user interface. The received gesture can be received by the Task Management Module 540 and sent to the Gesture Module 576 of the Multi-Display Management Module 524. The Gesture Module 576 may interpret the gesture as wanting to open a window associated with the executed application, as described in conjunction with Figs. 7F through 7K. In embodiments, the Task Management Module 540 places the user interface interaction in the task stack 552 to be acted upon by the Multi-Display Management Module 524. Further, the Task Management Module 540 waits for information from the Multi-Display Management Module 524 to send instructions to the Window Management Module 532 to create the window in the window stack.

[0123] The Multi-Display Management Module 524, upon receiving instruction from the Task Management Module 540, determines to which portion of the composite display 760, the opened window should be associated, in step 912. In embodiments, the window is either a full screen window displayed in LMax or PMax or a set of windows having a parent and child window displayed on different touch sensitive displays 110,114. For example, window 9 750 is associated with both touch sensitive displays 110,114, as shown in Fig. 7J. The opening of the window(s) 9 750, as described in conjunction with Figs. 7J through 7M, includes setting the association of window 9 750 to both touch sensitive displays 110,114, which comprises the entire portion of the composite display 764, as shown in Fig. 7J. In Figs. 7F through 7I, window 9.1 774 is associated with the left side of the composite display 764 and window 9.2 776 is

associated with the right side of the composite display 764. Thus, window 9.1 774 is associated with an identifier for the primary touch sensitive display 110 and window 9.2 776 is associated with an identifier for the secondary touch sensitive display 114.

[0124] In embodiments, the device state module 574 of the Multi-Display Management Module 524 may determine how the device is oriented or in what state the device is in, e.g., open, closed, portrait, etc. Further, the preferences module 572 and/or requirements module 580 may determine how the window is to be displayed. The gesture module 576 may determine the user's intentions about how the window is to be opened based on the type of gesture and the location of where the gesture is made.

[0125] The Display Configuration Module 568 may then use the input from the device state module 574, preferences module 572, and/or gesture module 576 to evaluate the current window stack 760 to determine the best place and the best dimensions, based on a visibility algorithm, to move the window. Thus, the Display Configuration Module 568 determines the best place to put the opened window at the top of the window stack 760, in step 916. In embodiments, the opened window is placed over substantially all of the composite display. The visibility algorithm, in embodiments, determines for all portions of the composite display, which windows are at the top of the stack. For example, the visibility algorithm determines that, after window 9 750 is opened, only window 9 750 is at the top of the stack 760, as viewed in Figs. 7J through 7M. In another example, the visibility algorithm determines that, after window 9.1 774 and window 9.2 776 are opened, only window 9.1 774 and window 9.2 776 are at the top of the stack 760, as viewed in Figs. 7F through 7I. Upon determining where to open the window(s), the Display Configuration Module 568 can change the display identifiers 816, dimensions 808, and/or stack position identifiers 812 for the one or more opened windows. The Multi-Display Management Module 524 can then send the display identifiers 816, dimensions 808, and/or stack position identifiers 812 back to the Task Management Module 540.

[0126] In embodiments, the Task Management Module 540 sends the display identifiers 816, dimensions 808, stack position identifiers 812, and/or other information and instructions to render the window(s) to the Window Management Module 532. The Window Management Module 532 and the Task Management Module 540 can change the logical data structure(s) 800, in step 924. Both the Task Management Module 540 and the Window Management Module 532 may manage copies of the window stack. These copies of the window stack can be synchronized

or kept similar through communications between the Window Management Module 532 and the Task Management Module 540. Thus, the Window Management Module 532 and the Task Management Module 540, based on the information determined by the Multi-Display Management Module 524, can change display identifiers 816, dimensions 808, and/or stack position identifiers 812 for the window(s). The logical data structure 800 may then be stored by both the Window Management Module 532 and the Task Management Module 540. Further, the Window Management Module 532 and the Task Management Module 540 may thereinafter manage the window stack and the logical data structure(s) 800.

[0127] The exemplary systems and methods of this disclosure have been described in relation to a window stack associated with a multi-screen device. However, to avoid unnecessarily obscuring the present disclosure, the preceding description omits a number of known structures and devices. This omission is not to be construed as a limitation of the scopes of the claims. Specific details are set forth to provide an understanding of the present disclosure. It should however be appreciated that the present disclosure may be practiced in a variety of ways beyond the specific detail set forth herein.

[0128] Furthermore, while the exemplary aspects, embodiments, and/or configurations illustrated herein show the various components of the system collocated, certain components of the system can be located remotely, at distant portions of a distributed network, such as a LAN and/or the Internet, or within a dedicated system. Thus, it should be appreciated, that the components of the system can be combined in to one or more devices, such as the software modules of Figs. 5A and 5B, or collocated on a particular node of a distributed network, such as an analog and/or digital telecommunications network, a packet-switch network, or a circuit-switched network. It will be appreciated from the preceding description, and for reasons of computational efficiency, that the components of the system can be arranged at any location within a distributed network of components without affecting the operation of the system. For example, the various components can be located in a switch such as a PBX and media server, gateway, in one or more communications devices, at one or more users' premises, or some combination thereof. Similarly, one or more functional portions of the system could be distributed between a telecommunications device(s) and an associated computing device.

[0129] Furthermore, it should be appreciated that the various links connecting the elements can be wired or wireless links, or any combination thereof, or any other known or later developed

element(s) that is capable of supplying and/or communicating data to and from the connected elements. These wired or wireless links can also be secure links and may be capable of communicating encrypted information. Transmission media used as links, for example, can be any suitable carrier for electrical signals, including coaxial cables, copper wire and fiber optics, and may take the form of acoustic or light waves, such as those generated during radio-wave and infra-red data communications.

[0130] Also, while the flowcharts have been discussed and illustrated in relation to a particular sequence of events, it should be appreciated that changes, additions, and omissions to this sequence can occur without materially affecting the operation of the disclosed embodiments, configuration, and aspects.

[0131] A number of variations and modifications of the disclosure can be used. It would be possible to provide for some features of the disclosure without providing others. For example in one alternative embodiment, the window stack 700 resembles a carousel or rolodex rather than a deck of cards. Thus, windows may circulate from one touch sensitive display 110 to another touch sensitive display 114. For example, window 6 724 may be push to the right and end up at the end of stack 764 behind window 716. If the stack continues to move to the right, eventually window 6 724 would appear on the second touch sensitive display 114 even though the window 724 was opened in the first touch sensitive display 110. These movements and changes in the stacks may be managed using the methods and logical data structures discussed above. In another alternative embodiment, there may be other arrangement for the window stack.

[0132] In yet another embodiment, the systems and methods of this disclosure can be implemented in conjunction with a special purpose computer, a programmed microprocessor or microcontroller and peripheral integrated circuit element(s), an ASIC or other integrated circuit, a digital signal processor, a hard-wired electronic or logic circuit such as discrete element circuit, a programmable logic device or gate array such as PLD, PLA, FPGA, PAL, special purpose computer, any comparable means, or the like. In general, any device(s) or means capable of implementing the methodology illustrated herein can be used to implement the various aspects of this disclosure. Exemplary hardware that can be used for the disclosed embodiments, configurations and aspects includes computers, handheld devices, telephones (e.g., cellular, Internet enabled, digital, analog, hybrids, and others), and other hardware known in the art. Some of these devices include processors (e.g., a single or multiple microprocessors), memory,

nonvolatile storage, input devices, and output devices. Furthermore, alternative software implementations including, but not limited to, distributed processing or component/object distributed processing, parallel processing, or virtual machine processing can also be constructed to implement the methods described herein.

[0133] In yet another embodiment, the disclosed methods may be readily implemented in conjunction with software using object or object-oriented software development environments that provide portable source code that can be used on a variety of computer or workstation platforms. Alternatively, the disclosed system may be implemented partially or fully in hardware using standard logic circuits or VLSI design. Whether software or hardware is used to implement the systems in accordance with this disclosure is dependent on the speed and/or efficiency requirements of the system, the particular function, and the particular software or hardware systems or microprocessor or microcomputer systems being utilized.

[0134] In yet another embodiment, the disclosed methods may be partially implemented in software that can be stored on a storage medium, executed on programmed general-purpose computer with the cooperation of a controller and memory, a special purpose computer, a microprocessor, or the like. In these instances, the systems and methods of this disclosure can be implemented as program embedded on personal computer such as an applet, JAVA® or CGI script, as a resource residing on a server or computer workstation, as a routine embedded in a dedicated measurement system, system component, or the like. The system can also be implemented by physically incorporating the system and/or method into a software and/or hardware system.

[0135] Although the present disclosure describes components and functions implemented in the aspects, embodiments, and/or configurations with reference to particular standards and protocols, the aspects, embodiments, and/or configurations are not limited to such standards and protocols. Other similar standards and protocols not mentioned herein are in existence and are considered to be included in the present disclosure. Moreover, the standards and protocols mentioned herein and other similar standards and protocols not mentioned herein are periodically superseded by faster or more effective equivalents having essentially the same functions. Such replacement standards and protocols having the same functions are considered equivalents included in the present disclosure.

[0136] The present disclosure, in various aspects, embodiments, and/or configurations, includes components, methods, processes, systems and/or apparatus substantially as depicted and described herein, including various aspects, embodiments, configurations embodiments, subcombinations, and/or subsets thereof. Those of skill in the art will understand how to make and use the disclosed aspects, embodiments, and/or configurations after understanding the present disclosure. The present disclosure, in various aspects, embodiments, and/or configurations, includes providing devices and processes in the absence of items not depicted and/or described herein or in various aspects, embodiments, and/or configurations hereof, including in the absence of such items as may have been used in previous devices or processes, e.g., for improving performance, achieving ease and\or reducing cost of implementation.

[0137] The foregoing discussion has been presented for purposes of illustration and description. The foregoing is not intended to limit the disclosure to the form or forms disclosed herein. In the foregoing Detailed Description for example, various features of the disclosure are grouped together in one or more aspects, embodiments, and/or configurations for the purpose of streamlining the disclosure. The features of the aspects, embodiments, and/or configurations of the disclosure may be combined in alternate aspects, embodiments, and/or configurations other than those discussed above. This method of disclosure is not to be interpreted as reflecting an intention that the claims require more features than are expressly recited in each claim. Rather, as the following claims reflect, inventive aspects lie in less than all features of a single foregoing disclosed aspect, embodiment, and/or configuration. Thus, the following claims are hereby incorporated into this Detailed Description, with each claim standing on its own as a separate preferred embodiment of the disclosure.

[0138] Moreover, though the description has included description of one or more aspects, embodiments, and/or configurations and certain variations and modifications, other variations, combinations, and modifications are within the scope of the disclosure, e.g., as may be within the skill and knowledge of those in the art, after understanding the present disclosure. It is intended to obtain rights which include alternative aspects, embodiments, and/or configurations to the extent permitted, including alternate, interchangeable and/or equivalent structures, functions, ranges or steps to those claimed, whether or not such alternate, interchangeable and/or equivalent structures, functions, ranges or steps are disclosed herein, and without intending to publicly dedicate any patentable subject matter.

What is claimed is:

1. A computer readable medium, having stored thereon, computer-executable instructions executable by a processor, the computer-executable instructions causing the processor to execute a method for managing a window stack for a device, the computer-executable instructions comprising:

instructions to receive a gesture to open a full screen window associated with the device, wherein a full screen window is operable to be displayed across two or more displays;

instructions to determine a display associated with the opened window;

instructions to determine a position in the window stack for the opened window; and

after determining the position of and the display associated with the opened window, instructions to create a logic data structure associated with the opened window to describe the position of the opened window in the window stack.

2. The computer readable medium as defined in claim 1, wherein the window stack is a logical arrangement of active and inactive windows for the device.

3. The computer readable medium as defined in claim 2, wherein the opened window is displayed on at least two displays of the device.

4. The computer readable medium as defined in claim 3, wherein the window is opened across substantially all of a composite display.

5. The computer readable medium as defined in claim 4, wherein the composite display occupies substantially all of the at least two displays.

6. The computer readable medium as defined in claim 5, wherein the logical data structure comprises one or more of:

a window identifier adapted to identify the opened window in relation to other windows in the window stack;

a window stack position identifier adapted to identify the position in the window stack for the opened window; and

a display identifier adapted to identify the composite display of the device, which the opened window is associated.

7. The computer readable medium as defined in claim 6, wherein a first display of the device is associated with a first portion of the stack and a second display of the device is associated with a second portion of the stack.

8. The computer readable medium as defined in claim 7, wherein the logic data structure further comprises a window stack identifier adapted to identify which both the first portion and second portion of which the opened window is associated.

9. A device comprising:

at least two displays;

a memory;

a processor in communication with the memory and each of the at least two displays, the processor operable to:

provide a composite display, wherein the composite display includes the at least two displays;

receive a gesture on the device, wherein the gesture opens at least one window that occupies a substantial portion of the composite display;

associate the at least one opened window with the composite display; and

after associating the at least one opened window with the composite display, create a logic data structure associated with the at least one opened window that describes the position of the at least one opened window as being on top of the window stack and displayed across the at least two displays.

10. The device as defined in claim 9, wherein at least one other window is altered by the opening of the at least one opened window.

11. The device as defined in claim **Error! Reference source not found.**, further comprising altering a second logic data structure associated with an altered window.

12. The device as defined in claim 9, wherein the at least one opened window is associated with all of the at least two displays.

13. The device as defined in claim 12, wherein the logic data structure comprises one or more of:

a window identifier adapted to identify the at least one opened window in relation to other windows in the window stack;

a window stack position identifier adapted to identify the position in the window stack for the at least one opened window; and

at least two display identifiers adapted to identify each of the at least two displays of the device which the at least one opened window is associated.

14. The device as defined in claim 13, wherein the at least one opened window includes a parent window and a child window.

15. The device as defined in claim 14, wherein the parent window is associated with a first portion of the composite display and the child window is associated with a second portion of the composite display.

16. A method for presenting a display for a device, the method comprising:
providing a composite display that spans at least a portion of two or more screens;
displaying a first window in at least one portion of the composite display
receiving a first gesture associated with opening a window;
interpreting the gesture as opening a full screen window; and
presenting the full screen window across the composite display spanning at least the portion of the two or more screens, wherein the full screen obscures the first window.

17. The method defined in claim 16, further comprising:
receiving a second gesture;
interpreting the second gesture as moving the full screen window; and
presenting the first window as obscuring at least a portion of the full screen window.

18. The method defined in claim 17, wherein presenting the full screen window appears to cover the first window.

19. The method defined in claim 18, wherein the second gesture is a flick gesture in a gesture capture region, and wherein the first window appears to cover the full screen window.

20. The method defined in claim 18, wherein the second gesture is a drag gesture in a gesture capture region, and wherein the first window appears to slide over the full screen window.

ABSTRACT

Systems and methods are provides for opening a full screen window in a window stack for a multi-screen device. The window stack can change based on the opening of a window. The system can receive a gesture indicating an application with a new window is to be executed or a new window is to be opened in the device. Upon receiving the gesture, the system determines that the new window is to occupy substantially all of a composite display that spans substantially all of the two screens of the device. Then, the system can determine that the full screen window is to be associated with the composite display and create a logic data structure associated with the opened window to describe the position of the opened window in the window stack.

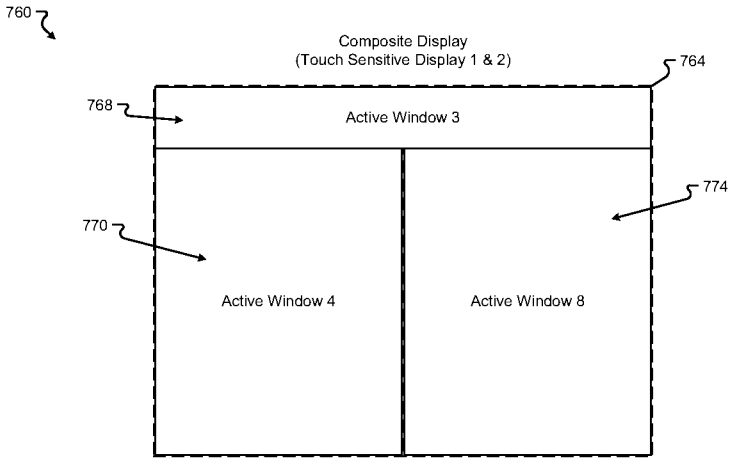


FIG. 7C

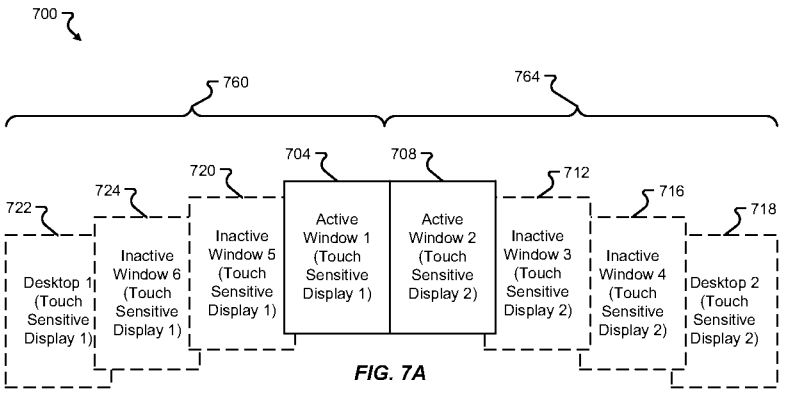


FIG. 7A

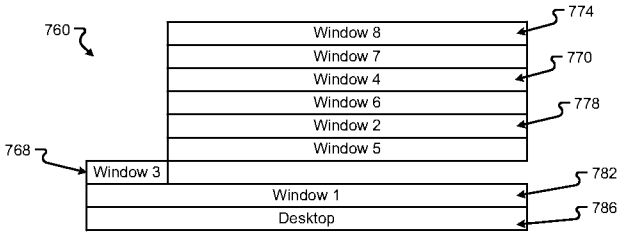


FIG. 7D

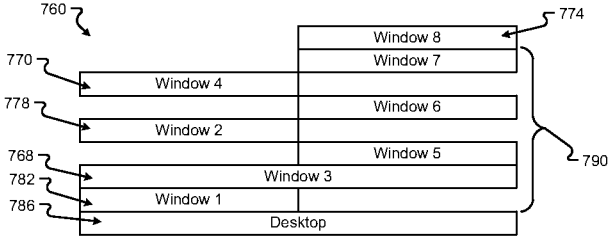


FIG. 7E

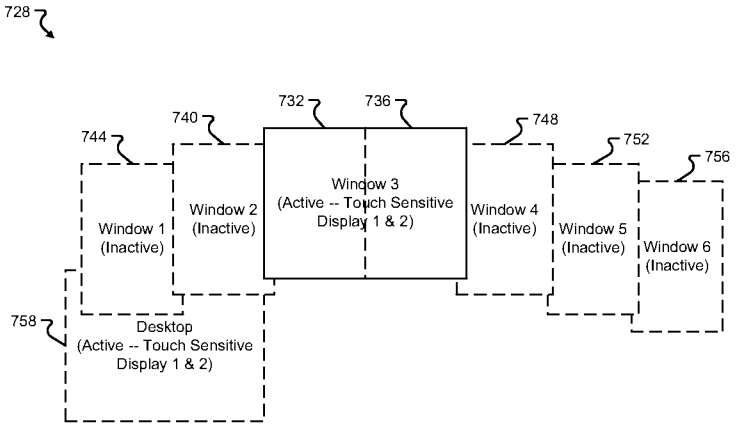


FIG. 7B

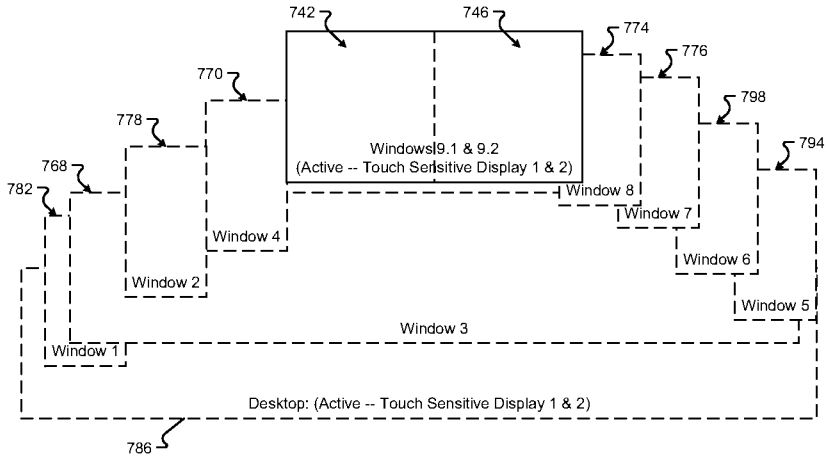


FIG. 7I

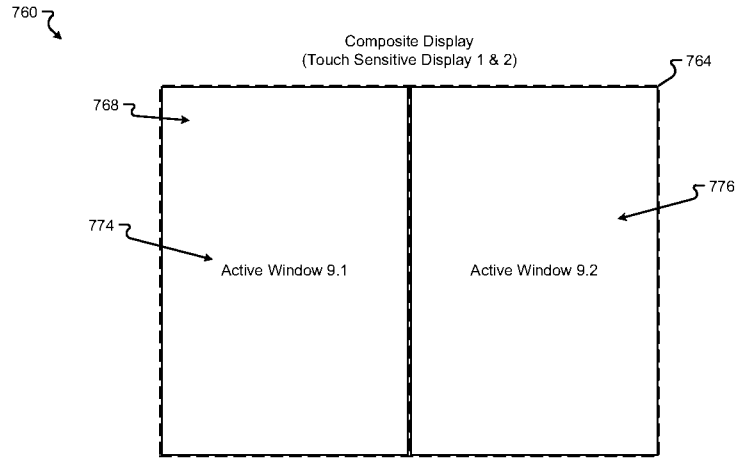


FIG. 7F

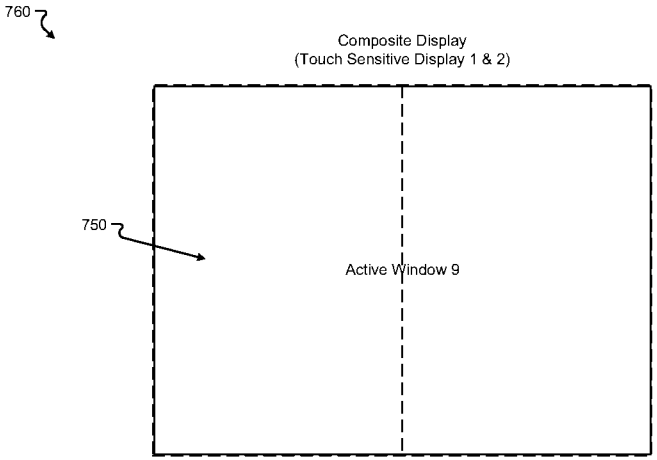


FIG. 7J

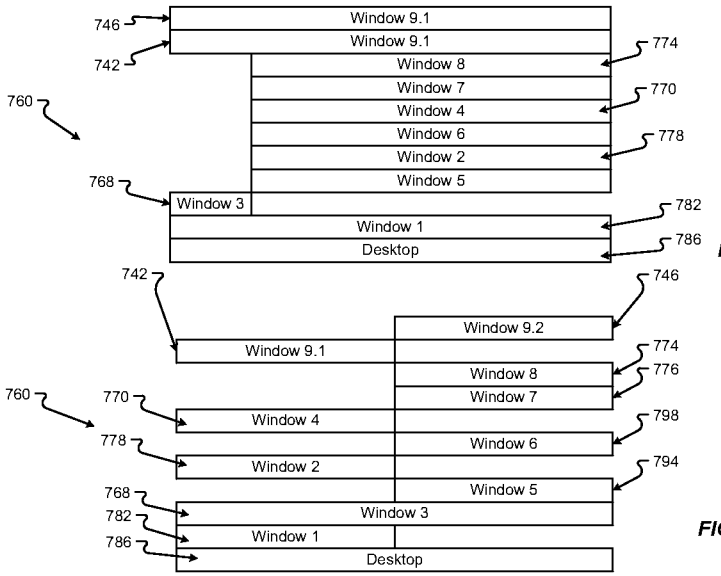


FIG. 7G

FIG. 7H

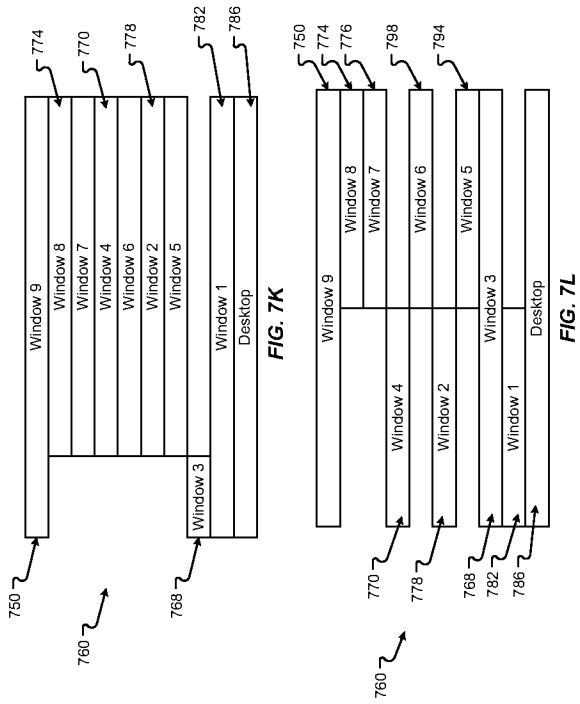


FIG. 7K

FIG. 7L

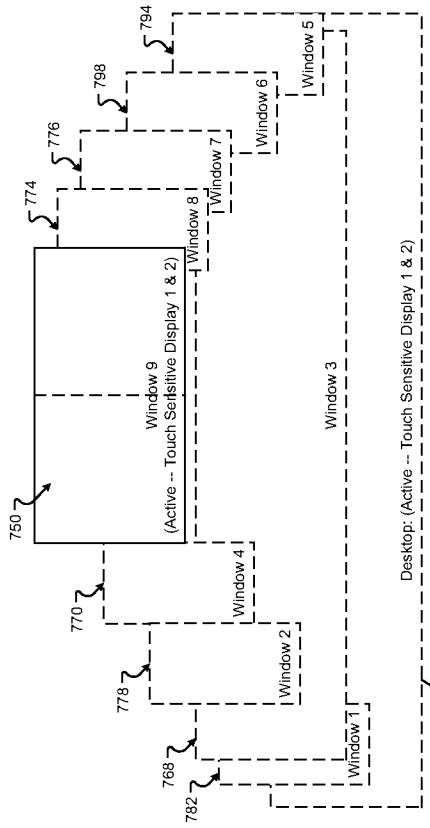


FIG. 7M

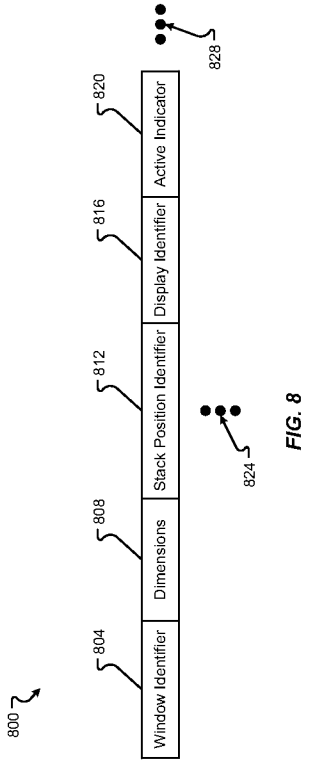


FIG. 8

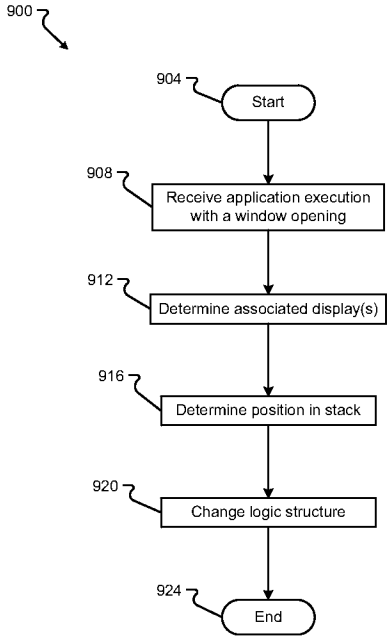


FIG. 9

WINDOW STACK MODIFICATION IN RESPONSE TO ORIENTATION CHANGE

[0100] Display Controls:

[0101] The device 100 manages desktops and/or windows with at least one window stack 700, 728, as shown in Figs. 7A and 7B. A window stack 700, 728 is a logical arrangement of active and/or inactive windows for a multi-screen device. For example, the window stack 700, 728 may be logically similar to a deck of cards, where one or more windows or desktops are arranged in order, as shown in Figs. 7A and 7B. An active window is a window that is currently being displayed on at least one of the touch sensitive displays 110, 114. For example, windows 704 and 708 are active windows and are displayed on touch sensitive displays 110 and 114. An inactive window is a window that was opened and displayed but is now "behind" an active window and not being displayed. In embodiments, an inactive window may be for an application that is suspended, and thus, the window is not displaying active content. For example, windows 712, 716, 720, and 724 are inactive windows

[0102] A window stack 700, 728 may have various arrangements or organizational structures. In the embodiment shown in Fig. 7A, the device 100 includes a first stack 760 associated with a first touch sensitive display 110 and a second stack associated with a second touch sensitive display 114. Thus, each touch sensitive display 110, 114 can have an associated window stack 760, 764. These two window stacks 760, 764 may have different numbers of windows arranged in the respective stacks 760, 764. Further, the two window stacks 760, 764 can also be identified differently and managed separately. Thus, the first window stack 760 can be arranged in order from a first window 704 to a next window 720 to a last window 724 and finally to a desktop 722, which, in embodiments, is at the "bottom" of the window stack 760. Likewise, the second stack 764 can be arranged from a first window 708 to a next window 712 to a last window 716, and finally to a desktop 718, which, in embodiments, is a single desktop area, with desktop 722, under all the windows in both window stack 760 and window stack 764. A logical data structure for managing the two window stacks 760, 764 may be as described in conjunction with Fig. 8.

[0103] Another arrangement for a window stack 728 is shown in Fig. 7B. In this embodiment, there is a single window stack 728 for both touch sensitive displays 110, 114. Thus, the window stack 728 is arranged from a desktop 758 to a first window 744 to a last window 756. A window can be arranged in a position among all windows without an association to a specific touch sensitive display 110, 114. In this embodiment, a window is in the order of windows. Further, at

least one window is identified as being active. For example, a single window may be rendered in two portions 732 and 736 that are displayed on the first touch sensitive screen 110 and the second touch sensitive screen 114. The single window may only occupy a single position in the window stack 728 although it is displayed on both displays 110, 114.

[0104] Yet another arrangement of a window stack 760 is shown in Figs. 7C through 7E. The window stack 760 is shown in a plan view (Fig. 7C) and two "elevation" views (Figs. 7D and 7E). In Fig. 7C, the top of the window stack 760 is shown. Two sides of the window stack 760 are shown in Figs. 7D and 7E. In this embodiment, the window stack 760 resembles a stack of bricks. The windows are stacked on each other. Looking from the top of the window stack 760 in Fig, 7C, only the top most windows in the window stack 760 are seen in different portions of the composite display 764. The composite display 764 represents a logical model for the entire display area of the device 100, which can include touch sensitive display 110 and touch sensitive display 114. A desktop 786 or a window can occupy part or all of the composite display 764.

[0105] In the embodiment shown, the desktop 786 is the lowest display or "brick" in the window stack 760. Thereupon, window 1 782, window 2 778, window 3 768, and window 4 770 are layered. Window 1 782, window 3 768, window 2 778, and window 4 770 only occupy a portion of the composite display 764. Thus, another part of the stack 760 includes window 8 774 and windows 5 through 7 shown in section 790. Only the top window in any portion of the composite display 764 is actually rendered and displayed. Thus, as shown in the top view in Fig. 7C, window 4 770, window 8 774, and window 3 768 are displayed as being at the top of the display in different portions of the window stack 760. A window can be dimensioned to occupy only a portion of the composite display 760 to "reveal" windows lower in the window stack 760. For example, window 3 768 is lower in the stack than both window 4 770 and window 8 774 but is still displayed. A logical data structure to manage the window stack can be as described in conjunction with Fig. 8.

[0106] When a new window is opened, the newly activated window is generally positioned at the top of the stack. However, where and how the window is positioned within the stack can be a function of the orientation of the device 100, the context of what programs, functions, software, etc. are being executed on the device 100, how the stack is positioned when the new window is opened, etc. To insert the window in the stack, the position in the stack for the window is determined and the touch sensitive display 110, 114 to which the window is associated may also

be determined. With this information, a logical data structure for the window can be created and stored. When user interface or other events or tasks change the arrangement of windows, the window stack(s) can be changed to reflect the change in arrangement. It should be noted that these same concepts described above can be used to manage the one or more desktops for the device 100.

[0107] A logical data structure 800 for managing the arrangement of windows or desktops in a window stack is shown in Fig. 8. The logical data structure 800 can be any data structure used to store data whether an object, record, file, etc. The logical data structure 800 can be stored in any type of database or data storage system, regardless of protocol or standard. In embodiments, the logical data structure 800 includes one or more portions, fields, attributes, etc. that store data in a logical arrangement that allows for easy storage and retrieval of the information. Hereinafrer, these one or more portions, fields, attributes, etc. shall be described simply as fields. The fields can store data for a window identifier 804, dimensions 808, a stack position identifier 812, a display identifier 816, and/or an active indicator 820. Each window in a window stack can have an associated logical data structure 800. While only a single logical data structure 800 is shown in Fig. 8, there may be more or fewer logical data structures 800 used with a window stack (based on the number of windows or desktops in the stack), as represented by ellipses 824. Further, there may be more or fewer fields than those shown in Fig. 8, as represented by ellipses 828.

[0108] A window identifier 804 can include any identifier (ID) that uniquely identifies the associated window in relation to other windows in the window stack. The window identifier 804 can be a globally unique identifier (GUID), a numeric ID, an alphanumeric ID, or other type of identifier. In embodiments, the window identifier 804 can be one, two, or any number of digits based on the number of windows that can be opened. In alternative embodiments, the size of the window identifier 804 may change based on the number of windows opened. While the window is open, the window identifier 804 may be static and remain unchanged.

[0109] Dimensions 808 can include dimensions for a window in the composite display 760. For example, the dimensions 808 can include coordinates for two or more corners of the window or may include one coordinate and dimensions for the width and height of the window. These dimensions 808 can delineate what portion of the composite display 760 the window may occupy, which may be the entire composite display 760 or only part of the composite display

760. For example, window 4 770 may have dimensions 808 that indicate that the window 770 will occupy only part of the display area for composite display 760, as shown in Figs. 7C through 7E. As windows are moved or inserted in the window stack, the dimensions 808 may change.

[0110] A stack position identifier 812 can be any identifier that can identify the position in the stack for the window or may be inferred from the window's control record within a data structure, such as a list or a stack. The stack position identifier 812 can be a GUID, a numeric ID, an alphanumeric ID, or other type of identifier. Each window or desktop can include a stack position identifier 812. For example, as shown in Fig. 7A, window 1 704 in stack 1 760 can have a stack position identifier 812 of 1 identifying that window 704 is the first window in the stack 760 and the active window. Similarly, window 6 724 can have a stack position identifier 812 of 3 representing that window 724 is the third window in the stack 760. Window 2 708 can also have a stack position identifier 812 of 1 representing that window 708 is the first window in the second stack 764. As shown in Fig. 7B, window 1 744 can have a stack position identifier 812 of 1, window 3, rendered in portions 732 and 736, can have a stack position identifier 812 of 3, and window 6 756 can have a stack position identifier 812 of 6. Thus, depending on the type of stack, the stack position identifier 812 can represent a window's location in the stack.

[0111] A display identifier 816 can identify that the window or desktop is associated with a particular display, such as the first display 110 or the second display 114, or the composite display 760 composed of both displays. While this display identifier 816 may not be needed for a multi-stack system, as shown in Fig. 7A, the display identifier 816 can indicate whether a window in the serial stack of Fig. 7B is displayed on a particular display. Thus, window 3 may have two portions 732 and 736 in Fig. 7B. The first portion 732 may have a display identifier 816 for the first display 110 while the second portion 736 may have a display identifier 816 for the second display 114. However, in alternative embodiments, the window may have two display identifiers 816 that represent that the window is displayed on both of the displays 110, 114, or a display identifier 816 identifying the composite display 760. In another alternate embodiment, the window may have a single display identifier 816 to represent that the window is displayed on both of the displays 110, 114.

[0112] Similar to the display identifier 816, an active indicator 820 may not be needed with the dual stack system of Fig. 7A, as the window in stack position 1 is active and displayed. In the

system of Fig. 7B, the active indicator 820 can indicate which window(s) in the stack is being displayed. Thus, window 3 may have two portions 732 and 736 in Fig. 7B. The first portion 732 may have an active indicator 820 while the second portion 736 may also have an active indicator 820. However, in alternative embodiments, window 3 may have a single active indicator 820. The active indicator 820 can be a simple flag or bit that represents that the window is active or displayed.

[0113] An embodiment of a method 900 for creating a window stack is shown in Fig. 9. While a general order for the steps of the method 900 is shown in Fig. 9. Generally, the method 900 starts with a start operation 904 and ends with an end operation 924. The method 900 can include more or fewer steps or can arrange the order of the steps differently than those shown in Fig. 9. The method 900 can be executed as a set of computer-executable instructions executed by a computer system and encoded or stored on a computer readable medium. Hereinafter, the method 900 shall be explained with reference to the systems, components, modules, software, data structures, user interfaces, etc. described in conjunction with Figs. 1-8.

[0114] A multi-screen device 100 can receive activation of a window, in step 908. In embodiments, the multi-screen device 100 can receive activation of a window by receiving an input from the touch sensitive display 110 or 114, the configurable area 112 or 116, a gesture capture region 120 or 124, or some other hardware sensor operable to receive user interface inputs. The processor may execute the Task Management Module 540 may receive the input. The Task Management Module 540 can interpret the input as requesting an application task to be executed that will open a window in the window stack.

[0115] In embodiments, the Task Management Module 540 places the user interface interaction in the task stack 552 to be acted upon by the Display Configuration Module 568 of the Multi-Display Management Module 524. Further, the Task Management Module 540 waits for information from the Multi-Display Management Module 524 to send instructions to the Window Management Module 532 to create the window in the window stack.

[0116] The Multi-Display Management Module 524, upon receiving instruction from the Task Management Module 540, determines to which touch portion of the composite display 760, the newly activated window should be associated, in step 912. For example, window 4 770 is associated with the a portion of the composite display 764 In embodiments, the Device State Module 574 of the Multi-Display Management Module 524 may determine how the device is

oriented or in what state the device is in, e.g., open, closed, portrait, etc. Further, the Preferences Module 572 and/or requirements module 580 may determine how the window is to be displayed. The gesture module 576 may determine the user's intentions about how the window is to be opened based on the type of gesture and the location of where the gesture is made.

[0117] The Display Configuration Module 568 may use the input from these modules and evaluate the current window stack 760 to determine the best place and the best dimensions, based on a visibility algorithm, to open the window. Thus, the Display Configuration Module 568 determines the best place to put the window at the top of the window stack 760, in step 916. Upon determining where to open the window, the Display Configuration Module 568 can assign a display identifier 816 and possibly dimensions 808 to the window. The display identifier 816 and possibly dimensions 808 can then be sent back to the Task Management Module 540. The Task Management Module 540 may then assign the window a stack position identifier 812 indicating the windows position in the window stack.

[0118] In embodiments, the Task Management Module 540 sends the window stack information and instructions to render the window to the Window Management Module 532. The Window Management Module 532 and the Task Management Module 540 can create the logical data structure 800, in step 924. Both the Task Management Module 540 and the Window Management Module 532 may create and manage copies of the window stack. These copies of the window stack can be synchronized or kept similar through communications between the Window Management Module 532 and the Task Management Module 540. Thus, the Window Management Module 532 and the Task Management Module 540, based on the information determined by the Multi-Display Management Module 524, can assign dimensions 808, a stack position identifier 812 (e.g., window 1 782, window 4 770, etc.), a display identifier 816 (e.g., touch sensitive display 1 110, touch sensitive display 2 114, composite display identifier, etc,), and an active indicator 820, which is generally always set when the window is at the "top" of the stack. The logical data structure 800 may then be stored by both the Window Management Module 532 and the Task Management Module 540. Further, the Window Management Module 532 and the Task Management Module 540 may thereinafter manage the window stack and the logical data structure(s) 800.

[0119] Fig. 10 shows a two-dimensional graphical representation of an embodiment of the present device 100 with logical window stack transitioning from a portrait configuration to a

landscape configuration. The device 100 is in an open portrait state 320, where two windows of the stack are currently being displayed on the primary screen 104 and the secondary screen 108, represented as surrounded by solid lines. Stack windows 1004, 1008, 1012, 1016 not currently displayed to the primary screen 104 and the secondary screen 104 are represented as surrounded by dashed lines. These windows may be accessed a number of ways, including providing directional user gestures input from the primary and secondary screens 104, 108. The device 100 is rotated counter-clockwise 1020 or clockwise 1024 to produce a specific logical stack configuration and corresponding user experience dependent upon the direction of the device's 100 rotation.

[0120] For example, afrer or while the device 100 is rotated, whether clockwise 1024 or counter-clockwise 1020, the Device State Module 574 of the Multi-Display Management Module 524 determines how to display the windows of the stack to the primary and secondary screens 104, 108. In accordance with at least one embodiment, the Device State Module 574 detects that the device 100 is rotated from a portrait configuration to a landscape configuration and as a result at least one of the Preferences Module 572 and the Requirements Module 580 determines to display the windows of the stack to one or more of the primary and secondary screens 104, 108 in a landscape orientation. Specifically the window displayed to the primary screen 104 is rotated from displaying in a first portrait configuration 604 to displaying in a first landscape configuration 612, while the window displayed to the secondary screen 108 is rotated from displaying in a second portrait configuration 608 to displaying in a second landscape configuration 616. After or while the device 100 is rotated, the dimensions of a displayed window may be recalculated to fit the new view configuration. As described, the determination of displaying a given window in either portrait or landscape mode may be dependent on device state, orientation, gesture or other input received, or in response to an output. In at least one embodiment it is anticipated that once a display orientation is determined, the device 100 may continue to display new, existing, or stack windows in that specific orientation until interrupted by a device state change, a user input, a response to: an output, a programmed condition, or other stimulus.

[0121] In accordance with the present disclosure, the user experience changes when the device 100 is rotated clockwise 1024 or counter-clockwise 1020 from a portrait configuration. As shown in Fig. 10, when the device 100 is rotated, the primary screen 104 displays data associated

with window 1in a landscape configuration while the secondary screen displays data associated with window 2 in a landscape configuration.

[0122] When the device 100 is rotated counter-clockwise 1020 from a portrait configuration, the primary screen 104 is oriented below the secondary screen 108. In an embodiment of the present disclosure, the window stack, or stacks, may be virtually reoriented in the same direction and manner as the primary and secondary screens 104, 108. Specifically, the stack windows associated with the primary screen 104 may be virtually oriented below the primary screen 104, while the stack windows associated with the secondary screen 108 may be virtually oriented above the secondary screen 108. This virtual organization of stack windows allows a user to intuitively access windows of the stack to display on the primary and/or secondary screen 104, 108 by, among other things, providing a directional user gesture input via one of the touch sensitive displays 110, 114. If the device 100 is rotated counter-clockwise 1020 from a portrait configuration as shown in Fig. 10, a user may access stack window 3 1004 by providing a directional user gesture input on the touch sensitive display 110 from any point on the display 110 in an upward general direction toward the secondary screen 108. In some embodiments this directional user gesture input may cross over more than one screen to achieve the same result. If a user desires access to stack window 4 1012 the user may provide a directional user gesture input on the touch sensitive display 112 from any point on the display 112 in a downward general direction toward the primary screen 104. As would be expected, the user may access other windows of the stack or previously displayed windows by repeating the user gesture input and/or by providing a prolonged and/or extended direction user gesture input.

[0123] When the device 100 is rotated clockwise 1024 from a portrait configuration, the primary screen 104 is oriented above the secondary screen 108. As stated above, the window stack, or stacks, may be virtually reoriented in the same direction and manner as the primary and secondary screens 104, 108. Specifically, the stack windows associated with the primary screen 104 may be virtually oriented above the primary screen 104, while the stack windows associated with the secondary screen 108 may be virtually oriented below the secondary screen 108. Therefore, if the device 100 is rotated clockwise 1024 from a portrait configuration as shown in Fig. 10, a user may access stack window 3 1004 by providing a directional user gesture input on the touch sensitive display 110 from any point on the display 110 in a downward general direction toward the secondary screen 108. In some embodiments this directional user gesture

input may cross over more than one screen to achieve the same result. If a user desires access to stack window 4 1012 the user may provide a directional user gesture input on the touch sensitive display 112 from any point on the display 112 in an upward general direction toward the primary screen 104.

[0124] An embodiment of a method 1100 for determining the window stack orientation and control is shown in Fig. 11. While a general order for the steps of the method 1100 is shown in Fig. 11, the method 1100 can include more or fewer steps or can arrange the order of the steps differently than those shown. The method 1100 begins with a start operation 1104 and ends with an end operation 1132. The method 1100 can be executed as a set of computer-executable instructions executed by a computer system and encoded or stored on a computer readable medium. Hereinafrer, the method 1100 shall be explained with reference to the systems, components, modules, software, data structures, user interfaces, etc. described in conjunction with Figs. 1-10.

[0125] A multi-screen device 100 can determine a change in orientation in step 1108. In embodiments, the multi-screen device 100 determines the change in orientation when the device 100 receives an input from the touch sensitive display 110 or 114, the configurable area 112 or 116, a gesture capture region 120 or 124, or some other sensor that is configured to detect a change in orientation state. Next, in optional step 1112, the device 100 determines the relative orientation of at least one of the screens. In accordance with the present disclosure, it is anticipated that this determination can be achieved using input received from a hardware sensor alone or in combination with variables and/or known constants. Examples of hardware sensors include accelerometers, gyroscopes, or other sensing elements. Variables may include changing device 100 state information such as a screen's display state information and whether a screen is oriented in a portrait 604, 608 or landscape 612, 616 state at any given moment. Known constants may include fixed screen locations (i.e., the primary screen 104 is always located to one side of the secondary screen 108), fixed screen dimensions, combinations thereof, etc. Specifically, in one embodiment, the device 100 may use information received from an accelerometer or other hardware sensor to detect the direction of the gravity vector and use the information in combination with known parameters to determine a screen's position. For example, if the gravity vector is pointing in the direction of the left side of a portrait oriented 604, 608 primary or secondary screen 104, 108, then the primary screen 104 is oriented below

the secondary screen 108 in a landscape state 612. However, if the gravity vector is pointing in the direction of the top side of the primary screen then the secondary screen 108 is to the left side of the primary screen 104 in a portrait state 608. In embodiments, this determination can be made using either or both screens 104, 108 of a device 100.

[0126] After or in lieu of determining the screen orientation, the window stack may be correspondingly sized and oriented to best suit the determined orientation of the device 100 and its screens (step 1116). The methods used to determine the orientation of the windows stack may be identical to the methods described above for determining a screen orientation. It is anticipated that the windows stack will be configured to display in the same orientation as that of a particular screen, that is, a device 100 with a landscape oriented screen may configure its window stack to display in a landscape orientation. In contrast, a device 100 with a portrait oriented screen may configure its window stack to display in a portrait orientation.

[0127] In embodiments, the user experience is dependent upon the specific orientation of the device 100. Because the device 100 may include the ability to logically arrange the windows stack associated with a specific orientation of the device 100, the user is able to more effectively navigate and utilize data accessed from the device 100. In some embodiments the device 100 detects user gesture input in the form of directional gesture input (step 1124). This directional gesture input may contain vector information that is be used by the device 100 to perform other functions. The vector information may include at least a direction and length of the input that the user initiated. In addition, the vector information could utilize speed of the gesture to control other functionality of the display. As a result, the directional gesture input used in conjunction with a window stack, that has been logically oriented and arranged, can control the display of data associated with a window or multiple windows of a windows stack. The direction component of the vector information may move a single window or plurality of windows from a window stack in the direction of the vector, while the length component of the vector information may be used to determine the number of windows to move or the magnitude of the move. Upon detecting the directional gesture input, the device 100 may control the display of a corresponding window or windows according to predetermined rules, responses, or conditions. The method is ended when a terminating input is received by the device 100 (at step 1132).

[0128] The exemplary systems and methods of this disclosure have been described in relation to a window stack associated with a multi-screen device. However, to avoid unnecessarily

obscuring the present disclosure, the preceding description omits a number of known structures and devices. This omission is not to be construed as a limitation of the scopes of the claims. Specific details are set forth to provide an understanding of the present disclosure. It should however be appreciated that the present disclosure may be practiced in a variety of ways beyond the specific detail set forth herein.

[0129] Furthermore, while the exemplary aspects, embodiments, and/or configurations illustrated herein show the various components of the system collocated, certain components of the system can be located remotely, at distant portions of a distributed network, such as a LAN and/or the Internet, or within a dedicated system. Thus, it should be appreciated, that the components of the system can be combined in to one or more devices, such as a _____, or collocated on a particular node of a distributed network, such as an analog and/or digital telecommunications network, a packet-switch network, or a circuit-switched network. It will be appreciated from the preceding description, and for reasons of computational efficiency, that the components of the system can be arranged at any location within a distributed network of components without affecting the operation of the system. For example, the various components can be located in a switch such as a PBX and media server, gateway, in one or more communications devices, at one or more users' premises, or some combination thereof. Similarly, one or more functional portions of the system could be distributed between a telecommunications device(s) and an associated computing device.

[0130] Furthermore, it should be appreciated that the various links connecting the elements can be wired or wireless links, or any combination thereof, or any other known or later developed element(s) that is capable of supplying and/or communicating data to and from the connected elements. These wired or wireless links can also be secure links and may be capable of communicating encrypted information. Transmission media used as links, for example, can be any suitable carrier for electrical signals, including coaxial cables, copper wire and fiber optics, and may take the form of acoustic or light waves, such as those generated during radio-wave and infra-red data communications.

[0131] Also, while the flowcharts have been discussed and illustrated in relation to a particular sequence of events, it should be appreciated that changes, additions, and omissions to this sequence can occur without materially affecting the operation of the disclosed embodiments, configuration, and aspects.

[0132] A number of variations and modifications of the disclosure can be used. It would be possible to provide for some features of the disclosure without providing others. For example in one alternative embodiment, the window stack 700 resembles a carousel or rolodex rather than a deck of cards. Thus, windows may circulate from one touch sensitive display 110 to another touch sensitive display 114. For example, window 6 724 may be pushed to the right and end up at the end of stack 764 behind window 716. If the stack continues to move to the right, eventually window 6 724 would appear on the second touch sensitive display 114 even though the window 724 was opened in the first touch sensitive display 110. In another alternative embodiment, there may be other arrangement for the window stack.

[0133] In yet another embodiment, the systems and methods of this disclosure can be implemented in conjunction with a special purpose computer, a programmed microprocessor or microcontroller and peripheral integrated circuit element(s), an ASIC or other integrated circuit, a digital signal processor, a hard-wired electronic or logic circuit such as discrete element circuit, a programmable logic device or gate array such as PLD, PLA, FPGA, PAL, special purpose computer, any comparable means, or the like. In general, any device(s) or means capable of implementing the methodology illustrated herein can be used to implement the various aspects of this disclosure. Exemplary hardware that can be used for the disclosed embodiments, configurations and aspects includes computers, handheld devices, telephones (e.g., cellular, Internet enabled, digital, analog, hybrids, and others), and other hardware known in the art. Some of these devices include processors (e.g., a single or multiple microprocessors), memory, nonvolatile storage, input devices, and output devices. Furthermore, alternative software implementations including, but not limited to, distributed processing or component/object distributed processing, parallel processing, or virtual machine processing can also be constructed to implement the methods described herein.

[0134] In yet another embodiment, the disclosed methods may be readily implemented in conjunction with software using object or object-oriented software development environments that provide portable source code that can be used on a variety of computer or workstation platforms. Alternatively, the disclosed system may be implemented partially or fully in hardware using standard logic circuits or VLSI design. Whether software or hardware is used to implement the systems in accordance with this disclosure is dependent on the speed and/or

efficiency requirements of the system, the particular function, and the particular software or hardware systems or microprocessor or microcomputer systems being utilized.

[0135] In yet another embodiment, the disclosed methods may be partially implemented in software that can be stored on a storage medium, executed on programmed general-purpose computer with the cooperation of a controller and memory, a special purpose computer, a microprocessor, or the like. In these instances, the systems and methods of this disclosure can be implemented as program embedded on personal computer such as an applet, JAVA® or CGI script, as a resource residing on a server or computer workstation, as a routine embedded in a dedicated measurement system, system component, or the like. The system can also be implemented by physically incorporating the system and/or method into a software and/or hardware system.

[0136] Although the present disclosure describes components and functions implemented in the aspects, embodiments, and/or configurations with reference to particular standards and protocols, the aspects, embodiments, and/or configurations are not limited to such standards and protocols. Other similar standards and protocols not mentioned herein are in existence and are considered to be included in the present disclosure. Moreover, the standards and protocols mentioned herein and other similar standards and protocols not mentioned herein are periodically superseded by faster or more effective equivalents having essentially the same functions. Such replacement standards and protocols having the same functions are considered equivalents included in the present disclosure.

[0137] The present disclosure, in various aspects, embodiments, and/or configurations, includes components, methods, processes, systems and/or apparatus substantially as depicted and described herein, including various aspects, embodiments, configurations embodiments, subcombinations, and/or subsets thereof. Those of skill in the art will understand how to make and use the disclosed aspects, embodiments, and/or configurations after understanding the present disclosure. The present disclosure, in various aspects, embodiments, and/or configurations, includes providing devices and processes in the absence of items not depicted and/or described herein or in various aspects, embodiments, and/or configurations hereof, including in the absence of such items as may have been used in previous devices or processes, e.g., for improving performance, achieving ease and\or reducing cost of implementation.

[0138] The foregoing discussion has been presented for purposes of illustration and description. The foregoing is not intended to limit the disclosure to the form or forms disclosed herein. In the foregoing Detailed Description for example, various features of the disclosure are grouped together in one or more aspects, embodiments, and/or configurations for the purpose of streamlining the disclosure. The features of the aspects, embodiments, and/or configurations of the disclosure may be combined in alternate aspects, embodiments, and/or configurations other than those discussed above. This method of disclosure is not to be interpreted as reflecting an intention that the claims require more features than are expressly recited in each claim. Rather, as the following claims reflect, inventive aspects lie in less than all features of a single foregoing disclosed aspect, embodiment, and/or configuration. Thus, the following claims are hereby incorporated into this Detailed Description, with each claim standing on its own as a separate preferred embodiment of the disclosure.

[0139] Moreover, though the description has included description of one or more aspects, embodiments, and/or configurations and certain variations and modifications, other variations, combinations, and modifications are within the scope of the disclosure, e.g., as may be within the skill and knowledge of those in the art, after understanding the present disclosure. It is intended to obtain rights which include alternative aspects, embodiments, and/or configurations to the extent permitted, including alternate, interchangeable and/or equivalent structures, functions, ranges or steps to those claimed, whether or not such alternate, interchangeable and/or equivalent structures, functions, ranges or steps are disclosed herein, and without intending to publicly dedicate any patentable subject matter.

What is claimed is:

1. A method of modifying a window stack in response to an orientation change of a multi-screen device, the method comprising:

determining that the multi-screen device has rotated from a first orientation to a second orientation;

determining a relative size and orientation of at least one window in a window stack stored on the multi-screen device based on the rotation of the multi-screen device; and

controlling the display of the at least one window stored in the window stack based on the determined relative size and orientation of the at least one window in the window stack.

2. The method of claim 1, wherein a first screen is determined to be above a second screen.

3. The method of claim 1, wherein a first screen is determined to be below a second screen.

4. The method of claim 1, wherein the rotation of the multi-screen device is determined, at least in part, based on data received from an accelerometer.

5. The method of claim 1, further comprising:

receiving a directional user gesture input on at least one of a first screen and a second screen to control a movement of the at least one window that is adjacent to the at least one of the first screen and second screen, wherein the directional user gesture input has both a direction and length; and

responding to the directional user gesture input with a command that moves the at least one window based, at least in part, on the direction of the directional user gesture input.

6. The method of claim 5, further comprising moving the at least one window based, at least in part, on the length of the directional user gesture input.

7. The method of claim 1, wherein the first orientation presents the first screen in a portrait state and the second orientation presents the first screen in a landscape state.

8. A non-transitory computer-readable medium having stored thereon instructions that cause a computing system to execute a method, the instructions comprising:

instructions configured to determine that the multi-screen device has rotated from a first orientation to a second orientation;

instructions configured to determine a relative size and orientation of at least one window in a window stack stored on the multi-screen device based on the based on the rotation of the multi-screen device; and

instructions configured to control the display of the at least one window stored in the window stack based on the determined relative size and orientation of the at least one window in the window stack.

9. The computer-readable medium of claim 8 further comprising:

instructions configured to receive a directional user gesture input on at least one of a first screen and a second screen to control a movement of the at least one window that is adjacent to the at least one of the first screen and second screen, wherein the directional user gesture input has both a direction and length; and

instructions configured to respond to the directional user gesture input with a command that moves the at least one window based, at least in part, on the direction of the directional user gesture input.

10. The computer-readable medium of claim 8, wherein the window stack is a logical arrangement of active and inactive windows for a multi-screen device.

11. The computer-readable medium of claim 8, wherein an active window is displayed on at least one of a first screen and a second screen of the multi-screen device.

12. The computer-readable medium of claim 8, wherein the relative size and orientation of the window stack can be associated with at least one of a first screen and a second screen of the multi-screen device.

13. The computer-readable medium as defined in claim 8, wherein the window stack is composed of a plurality of stacks and wherein a first screen is associated with a first stack in the plurality of stacks and a second screen is associated with a second stack in the plurality of stacks.

14. The computer-readable medium of claim 8, wherein a first screen is determined to be above a second screen.

15. The computer-readable medium of claim 8, wherein a first screen is determined to be below a second screen.

16. The computer-readable medium of claim 8, wherein the rotation of the multi-screen device is determined, at least in part, based on data received from an accelerometer.

17. The computer-readable medium of claim 9, further comprising moving the at least one window based, at least in part, on the length of the directional user gesture input.

18. A multi-screen user device, comprising:

a first screen including a first display area;

a second screen including a second display area;

a user gesture input area configured to receive physical input;

a first sensor configured to produce a first signal indicative of a relative orientation of a first screen to a second screen; and

the computer-readable medium of claim 8.

19. The device of claim 18, wherein the at least one of the first display area and second display area include the user gesture input area.

20. The device of claim 18, wherein the first screen is determined to be above the second screen, based at least partially on the first signal.

ABSTRACT

A multi-screen user device and methods for controlling data displayed thereby are disclosed. Specifically, the control of data displayed by at least one of the multiple screens of the multi-screen user device is conditioned upon the relative orientation of the multiple screens, whether the device orientation is changed from a first state to a second state, and user input received.

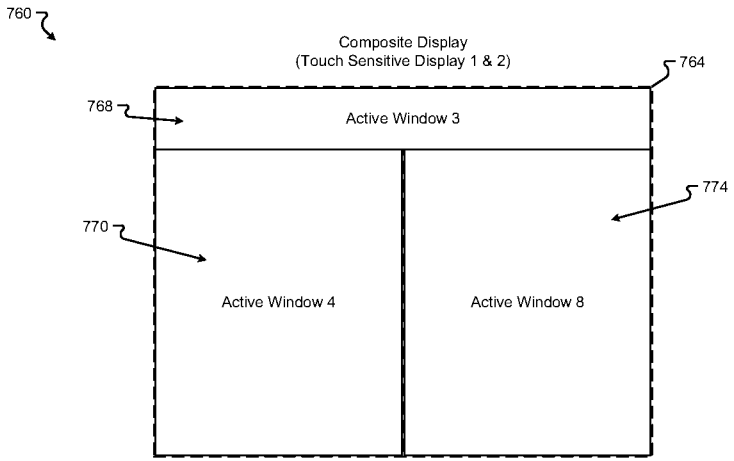


FIG. 7C

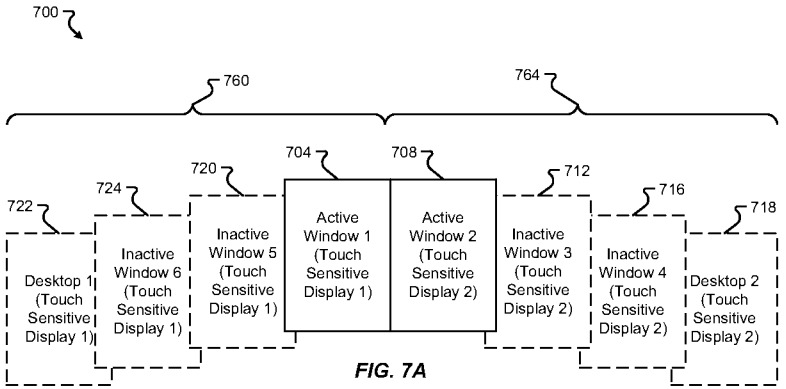


FIG. 7A

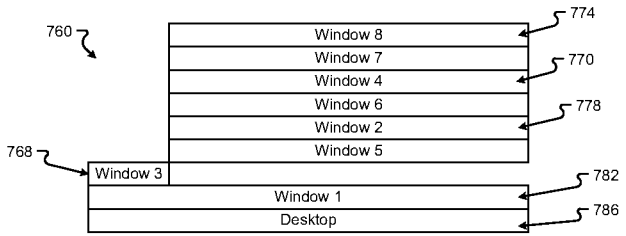


FIG. 7D

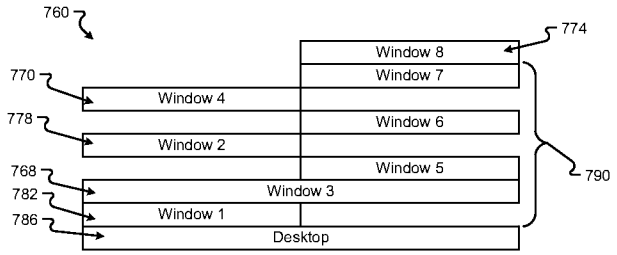


FIG. 7E

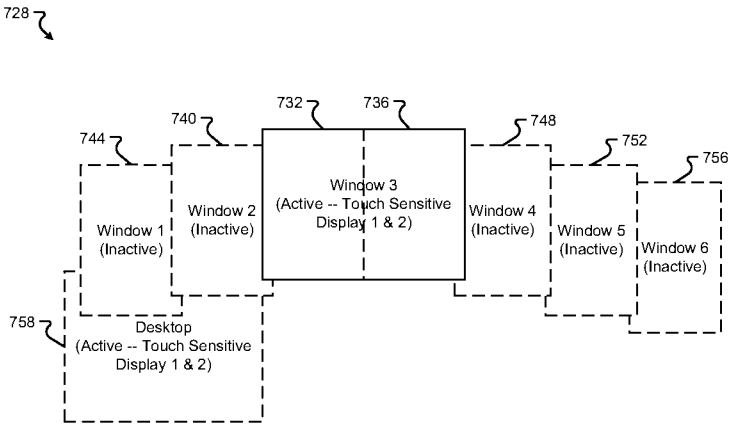


FIG. 7B

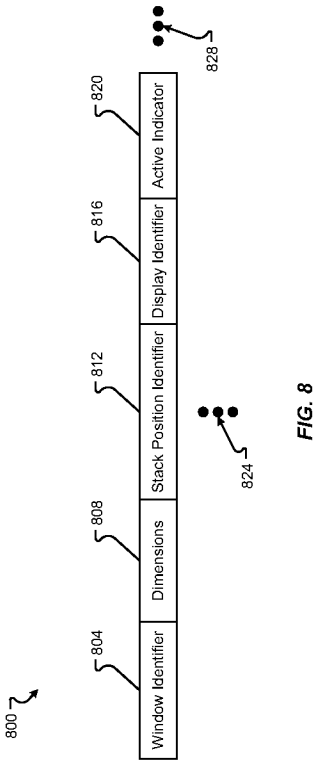


FIG. 8

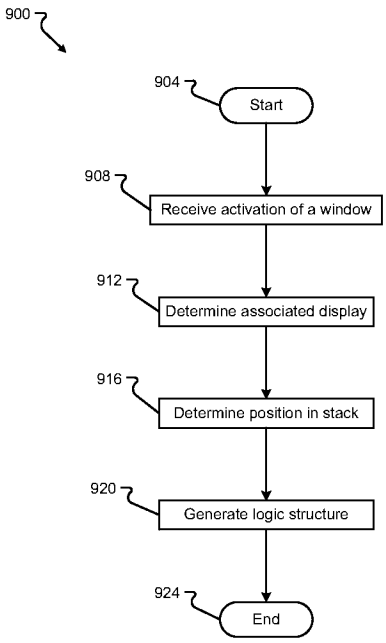


FIG. 9

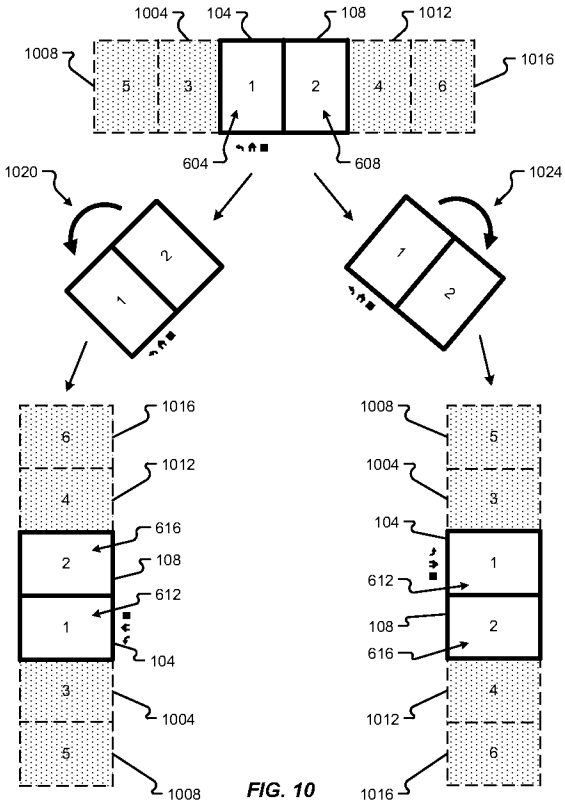


FIG. 10

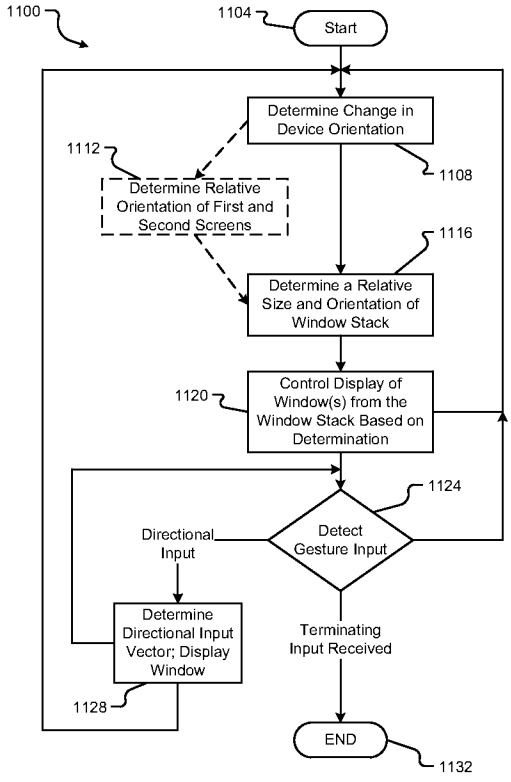


FIG. 11

[0100] The device 100 manages desktops and/or windows with at least one window stack 700, 728, as shown in Figs. 7A and 7B. A window stack 700, 728 is a logical arrangement of active and/or inactive windows for a multi-screen device. For example, the window stack 700, 728 may be logically similar to a deck of cards or a stack of bricks, where one or more windows or display objects (*e.g.*, desktops) are arranged in an order, as shown in Figs. 7A and 7B. An active window is a window that is currently being displayed on at least one of the touch sensitive displays 110, 114. For example, windows 104 and 108 are active windows and are displayed on touch sensitive displays 110 and 114. An inactive window is a window that was opened and displayed but is now "behind" an active window and not being displayed. In embodiments, an inactive window may be for an application that is suspended, and thus, the window is not displaying active content. For example, windows 712, 716, 720, and 724 are inactive windows.

[0101] A window stack 700, 728 may have various arrangements or organizational structures. In the embodiment shown in Fig. 7A, the device 100 includes a first stack 760 associated with a first touch sensitive display 110 and a second stack associated with a second touch sensitive display 114. Thus, each touch sensitive display 110, 114 can have an associated window stack 760, 764. In other embodiments, there is a single window stack that encompasses all of a composite display. The composite display is a logical structure that defines the entire display space including the two touch sensitive displays 110, 114. The device 100 can have a single stack for the composite display where the windows or display objects are sized to occupy a portion or all of the composite display.

[0102] In Fig. 7A, the two window stacks 760, 764 may have different numbers of windows or display objects arranged in the respective stacks 760, 764. Further, the two window stacks 760, 764 can also be identified differently and managed separately. Thus, the first window stack 760 can be arranged in order from a first window 704 to a next window 720 to a last window 724 and finally to a desktop 722, which, in embodiments, is at the "bottom" of the window stack 760. In embodiments, the desktop 722 is not always at the "bottom" as application windows can be arranged in the window stack below the desktop 722, and the desktop 722 can be brought to the "top" of a stack over other windows during a desktop reveal, as explained in conjunction with Figs. 7F through 7M. Likewise, the second stack 764 can be arranged from a first window 708 to a next window 712 to a last window 716, and finally to a desktop 718, which, in embodiments, is a single desktop area, with desktop 722 under all the windows in both window stack 760 and

window stack 764. A logical data structure for managing the two window stacks 760, 764 may be as described in conjunction with Fig. 8.

[0103] Another arrangement for a window stack 728 is shown in Fig. 7B. In this embodiment, there is a single window stack 728 for both touch sensitive displays 110, 114. Thus, the window stack 728 is arranged from a desktop 758 to a first window 744 to a last window 756. A window can be arranged in a position among all windows without an association to a specific touch sensitive display 110, 114. In this embodiment, the windows are ordered consecutively. Further, at least one window is identified as being active. For example, a single window may be rendered in the two portions 732 and 736 that are displayed on the first touch sensitive screen 110 and the second touch sensitive screen 114. The single window may only occupy a single position in the window stack 728 although it is displayed on both displays 110, 114.

[0104] Yet another arrangement of a window stack 760 is shown in Figs. 7C through 7E. The window stack 760 is shown in three "elevation" views. In Fig. 7C, the top of the window stack 760 is shown. Two adjacent sides of the window stack 760 are shown in Figs. 7D and 7E. In this embodiment, the window stack 760 resembles a stack of bricks. The windows are stacked on each other. Looking from the top of the window stack 760 in Fig, 7C, only the top most windows in the window stack 760 are seen in different portions of a composite display 764. A desktop 786 or a window can occupy part or all of the composite display 764.

[0105] In the embodiment shown, the desktop 786 is the lowest display or "brick" in the window stack 760. Thereupon, window 1 782, window 2 782, window 3 768, and window 4 770 are layered. Window 1 782, window 3 768, window 2 782, and window 4 770 only occupy a portion of the composite display 764. Thus, another part of the stack 760 includes window 8 774 and windows 5 through 7 shown in section 790. Only the top window in any portion of the composite display 764 is actually rendered and displayed. Thus, as shown in the top view in Fig. 7C, window 4 770, window 8 774, and window 3 768 are displayed by being at the top of the different portions of the window stack 760. A window can be dimensioned to occupy only a portion of the composite display 760 to "reveal" windows lower in the window stack 760. For example, window 3 768 is lower in the stack than both window 4 770 and window 8 774 but is still displayed.

[0106] Where and how the desktop is positioned within the stack can be a function of the orientation of the device 100, the context of what programs, functions, software, etc. are being

executed on the device 100, how the stack is positioned when the new desktop is revealed, etc. A logical data structure associated with the desktop can be changed when a desktop is revealed. When user interface or other events or tasks change the arrangement of the stack, the logical structures of the windows or desktop(s) can be changed to reflect the change in arrangement.

[0107] An embodiment of a window stack arrangement 760, which has changed because a desktop has been revealed, is shown in Figs. 7F through 7I. The window stack 760 is shown in a next state (from Figs. 7C through 7E) in four views, Figs. 7F through 7I. The top of the window stack 760 is shown in Fig. 7F, which is a view of what is displayed on the device 100. The two "sides" of the window stack 760 are shown in Figs. 7G and 7H. In this embodiment, the desktop has been revealed by a gesture. The gesture can be received from the touch sensitive display 110, 114, the configurable area 112,116, or from the gesture capture region 120, 124. In embodiments, a spread gesture 410 reveals the desktop 786. In alternative embodiments, the user presses the button 116c to reveal the desktop.

[0108] In interpreting the gesture to open the window(s) or execute the application, the Task Management Module 540 can request the Multi-Display Management Module 524 to determine to position the desktop 786 at the top of the stack. The Multi-Display Management Module 524 may then determine the dimensions of the desktop 786 to position the desktop 786 at the top of the stack 760 and displayed on both the first touch sensitive display 110 and the second touch sensitive display 110. Thus, looking from the top of the window stack 760 in Figs. 7C and 7F, desktop 786 is revealed on both the left and right side of the composite display 764.

[0109] Based on the opening of desktop 786, window 1 782 is the lowest window/display object or "brick" in the window stack 760. Thereupon, window 2 782, window 3 768, window 4 770, and the desktop 786 are layered on the lefr side of the composite display or lefr side of the stack 760. Window 1 782, window 3 768, window 2 782, and window 4 770 only occupy a portion of the composite display 764. Thus, another part the stack 760 includes window 5 794, window 6 798, window 7 776, window 8 774, and desktop 786 on the right side of the composite display 764 or the right side of the stack.

[0110] Only the top window/display object, in any portion of the composite display 764, is actually rendered and displayed. Thus, as shown in the top view in Fig. 7F, only the desktop 786 is displayed. The revealing of the desktop 786 covers window 8 774 window 4 770, and window

3 768. A window can be dimensioned to occupy only a portion of the composite display 760 to "reveal" windows lower in the window stack 760.

[0111] Fig. 7I provides a logical view, as would be understood by a user, of the window stack 760 afrer desktop 786 was revealed. Here, the desktop 786 appears to occupy both the first touch sensitive display 110 and the second touch sensitive display 114. The other windows would seem to be to the lefr or right of the desktop 786, as shown in Fig. 7I. Windo 3 768 and the desktop 786 would seem to extend across both displays. As such, if a user moved desktop 786 to the right, window 4 would appear to move onto the first touch sensitive display 110 from the lefr side of the first touch sensitive display 110. Likewise, if the user hid the desktop 786, window 8 774 would appear to move from the right and onto the second touch sensitive display 114. Thus, the window stack 760 maintains a "deck of cards" or a "film strip" type of structure for the user.

[0112] Still another embodiment of a window stack arrangement 760 that is changed because of a single window 774 is moved onto the desktop 786, is shown in Figs. 7J through 7M. The window stack 760 is shown in a next state (from Figs. 7F through 7I) in three views, Figs. 7J, 7K, and 7L. The top of the window stack 760 is shown in Fig. 7J, which is a view of what is displayed on the device 100. The two "sides" of the window stack 760 are shown in Figs. 7K and 7L. In this embodiment, window 8 774 is moved a gesture received from the touch sensitive display 110, 114 or from the gesture capture region 120, 124. In interpreting the gesture, the Task Management Module 540 can request the Multi-Display Management Module 524 to determine where to position window 9 750. The Multi-Display Management Module 524 may then determine the dimensions of window 9 750 to position the window at the top of the stack 760 and displayed across the first touch sensitive display 110 and the second touch sensitive display 114. Thus, looking from the side of the window stack 760 in Fig. 7K and 7L, desktop 9 750 opens over the right side of the composite display 764 and the lefr side of the composite display 764. Thus, desktop 9 750 covers window 8 774, window 4 770, and window 3 768.

[0113] Based on the opening if desktop 9 750, desktop 786 is the lowest window/display object or "brick" in the window stack 760. Thereupon, window 1 782, window 2 782, window 3 768, window 4 770, and a first portion of desktop 9 750 are layered on the right side of the stack 760. Window 1 782, window 3 768, window 2 782, and window 4 770 only occupy a portion of the composite display 764. Thus, another part of the stack 760 includes window 5 794, window 6 798, window 7 776, window 8 774, and a second portion of desktop 9.2 746 on the right side of

the right side of the stack 760. Thus, desktop 9 750 occupies both portions of the composite display 764. Only the top window in any portion of the composite display 764 is actually rendered and displayed. Thus, as shown in the top view in Fig. 7J, desktop 9 750 is displayed but on both the right and left side of the composite display 764. The opening of desktop 9 750 covered window 4 770, window 8 774, and window 3 768.

[0114] Fig. 7M provides a logical view, as would be understood by a user, of the window stack 760 after the desktop 9 750 was opened. Here, desktop 9 750 appears to occupy both the first touch sensitive display 110 and the second touch sensitive display 114. The other windows would seem to be to the left or right of the desktop 9 750, as shown in Fig. 7M. Window 3 768 and the desktop 786 would seem to extend across both displays. As such, if a user moved desktop 9 750 to the right, window 4 would appear to move onto the first touch sensitive display 110 ftom the left side of the first touch sensitive display 110. Likewise, if the user closed desktop 9 750, window 8 would appear to move ftom the right and onto the second touch sensitive display 114 while window 4 770 would appear to move onto the first touch sensitive display 110. Thus, the window stack 760 maintains a "deck of cards" type structure for the user. A logical data structure to manage the window stack can be as described in conjunction with Fig. 8.

[0115] A logical data structure 800 for managing the arrangement of windows or desktops in a window stack is shown in Fig. 8. The logical data structure 800 can be any data structure used to store data whether an object, record, file, etc. The logical data structure 800 can be stored in any type of database or data storage system, regardless of protocol or standard. In embodiments, the logical data structure 800 includes one or more portions, fields, attributes, etc. that store data in a logical arrangement that allows for easy storage and retrieval of the information. Hereinafter, these one or more portions, fields, attributes, etc. shall be described simply as fields. The fields can store data for a window identifier 804, dimensions 808, a stack position identifier 812, a display identifier 816, and/or an active indicator 820. Each window in a window stack can have an associated logical data structure 800. While only a single logical data structure 800 is shown in Fig. 8, there may be more or fewer logical data structures 800 used with a window stack (based on the number of windows or desktops in the stack), as represented by ellipses 824. Further, there may be more or fewer fields than those shown in Fig. 8, as represented by ellipses 828.

[0116] A window identifier 804 can include any identifier (ID) that uniquely identifies the associated window in relation to other windows in the window stack. The window identifier 804 can be a globally unique identifier (GUID), a numeric ID, an alphanumeric ID, or other type of identifier. In embodiments, the window identifier 804 can be one, two, or any number of digits based on the number of windows that can be opened. In alternative embodiments, the size of the window identifier 804 may change based on the number of windows opened. While the window is open, the window identifier 804 may be static and remain unchanged.

[0117] Dimensions 808 can include dimensions for a window in the composite display 760. For example, the dimensions 808 can include coordinates for two or more corners of the window or may include one coordinate and dimensions for the width and height of the window. These dimensions 808 can delineate what portion of the composite display 760 the window may occupy, which may the entire composite display 760 or only part of composite display 760. For example, window 4 770 may have dimensions 880 that indicate that the window 770 will occupy only part of the display area for composite display 760, as shown in Figs. 7c through 7E. As windows are moved or inserted in the window stack, the dimensions 808 may change.

[0118] A stack position identifier 812 can be any identifier that can identify the position in the stack for the window or may be inferred from the window's control record within a data structure, such as a list or a stack. The stack position identifier 812 can be a GUID, a numeric ID, an alphanumeric ID, or other type of identifier. Each window or desktop can include a stack position identifier 812. For example, as shown in Fig. 7A, window 1 704 in stack 1 760 can have a stack position identifier 812 of 1 identifying that window 704 is the first window in the stack 760 and the active window. Similarly, window 6 724 can have a stack position identifier 812 of 3 representing that window 724 is the third window in the stack 760. Window 2 708 can also have a stack position identifier 812 of 1 representing that window 708 is the first window in the second stack 764. As shown in Fig. 7B, window 1 744 can have a stack position identifier 812 of 1, window 3, rendered in portions 732 and 736, can have a stack position identifier 812 of 3, and window 6 756 can have a stack position identifier 812 of 6. Thus, depending on the type of stack, the stack position identifier 812 can represent a window's location in the stack.

[0119] A display identifier 816 can identify that the window or desktop is associated with a particular display, such as the first display 110 or the second display 114, or the composite display 760 composed of both displays. While this display identifier 816 may not be needed for

a multi-stack system, as shown in Fig. 7A, the display identifier 816 can indicate whether a window in the serial stack of Fig. 7B is displayed on a particular display. Thus, window 3 may have two portions 732 and 736 in Fig. 7B. The first portion 732 may have a display identifier 816 for the first display while the second portion 736 may have a display identifier 816 for the second display 114. However, in alternative embodiments, the window may have two display identifier 816 that represent that the window is displayed on both of the displays 110, 114, or a display identifier 816 identifying the composite display. In another alternate embodiment, the window may have a single display identifier 816 to represent that the window is displayed on both of the displays 110, 114.

[0120] Similar to the display identifier 816, an active indicator 820 may not be needed with the dual stack system of Fig. 7A, as the window in stack position 1 is active and displayed. In the system of Fig. 7B, the active indicator 820 can indicate which window(s) in the stack is being displayed. Thus, window 3 may have two portions 732 and 736 in Fig. 7. The first portion 732 may have an active indicator 820 while the second portion 736 may also have an active indicator 820. However, in alternative embodiments, window 3 may have a single active indicator 820. The active indicator 820 can be a simple flag or bit that represents that the window is active or displayed.

[0121] An embodiment of a method 900 for changing a window stack is shown in Fig. 9. While a general order for the steps of the method 900 is shown in Fig. 9. Generally, the method 900 starts with a start operation 904 and ends with an end operation 924. The method 900 can include more or fewer steps or can arrange the order of the steps differently than those shown in Fig. 9. The method 900 can be executed as a set of computer-executable instructions executed by a computer system and encoded or stored on a computer readable medium. Hereinafter, the method 900 shall be explained with reference to the systems, components, modules, software, data structures, user interfaces, etc. described in conjunction with Figs. 1-8.

[0122] A multi-screen device 100 can receive a gesture, as described in Figs. 4A-4H, directed to revealing a desktop 786, in step 908. The gesture can be a hardware input ftom a touch sensitive display 110, 114, ftom a configurable area 112,116, ftom a gesture capture region 120, 124, or some other user interface. The received gesture can be received by the Task Management Module 540 and sent to the Gesture Module 576 of the Multi-Display Management Module 524. The Gesture Module 576 may interpret the gesture as wanting to reveal a desktop,

as described in conjunction with Figs. 7F through 7M. In embodiments, the Task Management Module 540 places the user interface interaction in the task stack 552 to be acted upon by the Multi-Display Management Module 524. Further, the Task Management Module 540 waits for information from the Multi-Display Management Module 524 to send instructions to the Window Management Module 532 to create the window in the window stack 760.

[0123] The Multi-Display Management Module 524, upon receiving instruction from the Task Management Module 540, determines to which portion of the composite display 760, the desktop 786 should be associated, in step 912. In embodiments, the desktop 786 is a full screen display object displayed in LMax or PMax displayed on both touch sensitive displays 110,114. For example, desktop 786 is associated with both touch sensitive displays 110,114, as shown in Fig. 7F. When the desktop 786 was opened, the desktop 786 was associated with both touch sensitive displays 110,114, which comprises the entire portion of the composite display 764, as shown in Fig. 7F. Thus, desktop 786 is associated with an identifier for the primary touch sensitive display 110 and the secondary touch sensitive display 114.

[0124] In embodiments, the device state module 574 of the Multi-Display Management Module 524 may determine how the device is oriented or in what state the device is in, e.g., open, closed, portrait, etc. Further, the preferences module 572 and/or requirements module 580 may determine how the desktop 786 is to be displayed based on the preferences of the desktop 786. The gesture module 576 may determine the user's intentions about how the desktop 786 is to be revealed based on the type of gesture and the location of where the gesture is made.

[0125] The Display Configuration Module 568 may then use the input ftom the device state module 574, preferences module 572, and/or gesture module 576 to evaluate the current window stack 760. The desktop reveal may impact other windows in the window stack. The Display Configuration Module 568 may then determine to put the desktop 786 at the top of the window stack 760, in step 916. In embodiments, the desktop 786 is placed over substantially all of the composite display 764.

[0126] The visibility algorithm, in embodiments, determines for all portions of the composite display 764, which windows/display objects are at the top of the stack 760. For example, the visibility algorithm determines that, after the desktop 786 is revealed, only the desktop 786 is at the top of the stack 760, as viewed in Fig. 7F. In another example, the visibility algorithm determines that, after desktop 786 is revealed, only the desktop 786 is at the top of the stack 760,

as viewed in Figs. 7F through 7I. Upon determining where to reveal the desktop 786 or cover the other window(s), the Display Configuration Module 568 can change the display identifiers 816, dimensions 808, and/or stack position identifiers 812 for the desktop 786 and the one or more covered window(s). The Multi-Display Management Module 524 can then send the display identifiers 816, dimensions 808, and/or stack position identifiers 812 back to the Task Management Module 540.

[0127] In embodiments, the Task Management Module 540 sends the display identifiers 816, dimensions 808, stack position identifiers 812, and/or other information and instructions to render the desktop 786 to the Window Management Module 532. The Window Management Module 532 and the Task Management Module 540 can change the logical data structure(s) 800, in step 924. Both the Task Management Module 540 and the Window Management Module 532 may manage copies of the window stack. These copies of the window stack can be synchronized or kept similar through communications between the Window Management Module 532 and the Task Management Module 540. Thus, the Window Management Module 532 and the Task Management Module 540, based on the information determined by the Multi-Display Management Module 524, can change display identifiers 816, dimensions 808, and/or stack position identifiers 812 for the desktop 786 and the one or more window(s). The logical data structure 800 may then be stored by both the Window Management Module 532 and the Task Management Module 540. Further, the Window Management Module 532 and the Task Management Module 540 may thereinafrer manage the window stack and the logical data structure(s) 800.

[0128] The exemplary systems and methods of this disclosure have been described in relation to a window stack associated with a multi-screen device. However, to avoid unnecessarily obscuring the present disclosure, the preceding description omits a number of known structures and devices. This omission is not to be construed as a limitation of the scopes of the claims. Specific details are set forth to provide an understanding of the present disclosure. It should however be appreciated that the present disclosure may be practiced in a variety of ways beyond the specific detail set forth herein.

[0129] Furthermore, while the exemplary aspects, embodiments, and/or configurations illustrated herein show the various components of the system collocated, certain components of the system can be located remotely, at distant portions of a distributed network, such as a LAN

and/or the Internet, or within a dedicated system. Thus, it should be appreciated, that the components of the system can be combined in to one or more devices, such as the software modules of Figs. 5A and 5B, or collocated on a particular node of a distributed network, such as an analog and/or digital telecommunications network, a packet-switch network, or a circuit-switched network. It will be appreciated from the preceding description, and for reasons of computational efficiency, that the components of the system can be arranged at any location within a distributed network of components without affecting the operation of the system. For example, the various components can be located in a switch such as a PBX and media server, gateway, in one or more communications devices, at one or more users' premises, or some combination thereof. Similarly, one or more functional portions of the system could be distributed between a telecommunications device(s) and an associated computing device.

[0130] Furthermore, it should be appreciated that the various links connecting the elements can be wired or wireless links, or any combination thereof, or any other known or later developed element(s) that is capable of supplying and/or communicating data to and from the connected elements. These wired or wireless links can also be secure links and may be capable of communicating encrypted information. Transmission media used as links, for example, can be any suitable carrier for electrical signals, including coaxial cables, copper wire and fiber optics, and may take the form of acoustic or light waves, such as those generated during radio-wave and infra-red data communications.

[0131] Also, while the flowcharts have been discussed and illustrated in relation to a particular sequence of events, it should be appreciated that changes, additions, and omissions to this sequence can occur without materially affecting the operation of the disclosed embodiments, configuration, and aspects.

[0132] A number of variations and modifications of the disclosure can be used. It would be possible to provide for some features of the disclosure without providing others. For example in one alternative embodiment, the window stack 700 resembles a carousel or rolodex rather than a deck of cards. Thus, windows may circulate from one touch sensitive display 110 to another touch sensitive display 114. For example, window 6 724 may be push to the right and end up at the end of stack 764 behind window 716. If the stack continues to move to the right, eventually window 6 724 would appear on the second touch sensitive display 114 even though the window 724 was opened in the first touch sensitive display 110. These movements and changes in the

stacks may be managed using the methods and logical data structures discussed above. In another alternative embodiment, there may be other arrangement for the window stack.

[0133] In yet another embodiment, the systems and methods of this disclosure can be implemented in conjunction with a special purpose computer, a programmed microprocessor or microcontroller and peripheral integrated circuit element(s), an ASIC or other integrated circuit, a digital signal processor, a hard-wired electronic or logic circuit such as discrete element circuit, a programmable logic device or gate array such as PLD, PLA, FPGA, PAL, special purpose computer, any comparable means, or the like. In general, any device(s) or means capable of implementing the methodology illustrated herein can be used to implement the various aspects of this disclosure. Exemplary hardware that can be used for the disclosed embodiments, configurations and aspects includes computers, handheld devices, telephones (e.g., cellular, Internet enabled, digital, analog, hybrids, and others), and other hardware known in the art. Some of these devices include processors (e.g., a single or multiple microprocessors), memory, nonvolatile storage, input devices, and output devices. Furthermore, alternative software implementations including, but not limited to, distributed processing or component/object distributed processing, parallel processing, or virtual machine processing can also be constructed to implement the methods described herein.

[0134] In yet another embodiment, the disclosed methods may be readily implemented in conjunction with software using object or object-oriented software development environments that provide portable source code that can be used on a variety of computer or workstation platforms. Alternatively, the disclosed system may be implemented partially or fully in hardware using standard logic circuits or VLSI design. Whether software or hardware is used to implement the systems in accordance with this disclosure is dependent on the speed and/or efficiency requirements of the system, the particular function, and the particular software or hardware systems or microprocessor or microcomputer systems being utilized.

[0135] In yet another embodiment, the disclosed methods may be partially implemented in software that can be stored on a storage medium, executed on programmed general-purpose computer with the cooperation of a controller and memory, a special purpose computer, a microprocessor, or the like. In these instances, the systems and methods of this disclosure can be implemented as program embedded on personal computer such as an applet, JAVA® or CGI script, as a resource residing on a server or computer workstation, as a routine embedded in a

dedicated measurement system, system component, or the like. The system can also be implemented by physically incorporating the system and/or method into a software and/or hardware system.

[0136] Although the present disclosure describes components and functions implemented in the aspects, embodiments, and/or configurations with reference to particular standards and protocols, the aspects, embodiments, and/or configurations are not limited to such standards and protocols. Other similar standards and protocols not mentioned herein are in existence and are considered to be included in the present disclosure. Moreover, the standards and protocols mentioned herein and other similar standards and protocols not mentioned herein are periodically superseded by faster or more effective equivalents having essentially the same functions. Such replacement standards and protocols having the same functions are considered equivalents included in the present disclosure.

[0137] The present disclosure, in various aspects, embodiments, and/or configurations, includes components, methods, processes, systems and/or apparatus substantially as depicted and described herein, including various aspects, embodiments, configurations embodiments, subcombinations, and/or subsets thereof. Those of skill in the art will understand how to make and use the disclosed aspects, embodiments, and/or configurations afrer understanding the present disclosure. The present disclosure, in various aspects, embodiments, and/or configurations, includes providing devices and processes in the absence of items not depicted and/or described herein or in various aspects, embodiments, and/or configurations hereof, including in the absence of such items as may have been used in previous devices or processes, e.g., for improving performance, achieving ease and\or reducing cost of implementation.

[0138] The foregoing discussion has been presented for purposes of illustration and description. The foregoing is not intended to limit the disclosure to the form or forms disclosed herein. In the foregoing Detailed Description for example, various features of the disclosure are grouped together in one or more aspects, embodiments, and/or configurations for the purpose of streamlining the disclosure. The features of the aspects, embodiments, and/or configurations of the disclosure may be combined in alternate aspects, embodiments, and/or configurations other than those discussed above. This method of disclosure is not to be interpreted as reflecting an intention that the claims require more features than are expressly recited in each claim. Rather, as the following claims reflect, inventive aspects lie in less than all features of a single foregoing

disclosed aspect, embodiment, and/or configuration. Thus, the following claims are hereby incorporated into this Detailed Description, with each claim standing on its own as a separate preferred embodiment of the disclosure.

[0139] Moreover, though the description has included description of one or more aspects, embodiments, and/or configurations and certain variations and modifications, other variations, combinations, and modifications are within the scope of the disclosure, e.g., as may be within the skill and knowledge of those in the art, after understanding the present disclosure. It is intended to obtain rights which include alternative aspects, embodiments, and/or configurations to the extent permitted, including alternate, interchangeable and/or equivalent structures, functions, ranges or steps to those claimed, whether or not such alternate, interchangeable and/or equivalent structures, functions, ranges or steps are disclosed herein, and without intending to publicly dedicate any patentable subject matter.

What is claimed is:

1. A computer readable medium, having stored thereon, computer-executable instructions executable by a processor, the computer-executable instructions causing the processor to execute a method for managing a window stack for a device, the computer-executable instructions comprising:

instructions to receive a gesture to reveal a desktop associated with the device, wherein the desktop is operable to be displayed across two or more touch sensitive displays;

instructions to determine a second position in the window stack for the desktop, wherein the desktop is moved from a first position to a second position and wherein the second position is at a top of the window stack;

afrer determining the second position of desktop, instructions to change a logic data structure associated with the desktop to describe the second position of the desktop in the window stack; and

instructions to display the desktop to a user, wherein the desktop is displayed across the two or more touch sensitive displays.

2. The computer readable medium as defined in claim 1, wherein the window stack is a logical arrangement of active and inactive windows for the device.

3. The computer readable medium as defined in claim 2, wherein the window stack provides a representation of windows or display objects to the user.

4. The computer readable medium as defined in claim 3, wherein the across two or more touch sensitive displays is opened across substantially all of a composite display.

5. The computer readable medium as defined in claim 4, wherein the composite display occupies substantially all of the at least the two or more touch sensitive displays.

6. The computer readable medium as defined in claim 5, wherein the logical data structure comprises one or more of:

a window identifier adapted to identify the desktop in relation to other windows in the window stack;

a window stack position identifier adapted to identify the position in the window stack for the desktop; and

a display identifier adapted to identify the composite display of the device, which the desktop is associated.

7. The computer readable medium as defined in claim 6, wherein a first display of the device is associated with a first portion of the window stack and a second display of the device is associated with a second portion of the window stack.

8. The computer readable medium as defined in claim 7, wherein the logic data structure further comprises a window stack identifier adapted to identify both the first portion and second portion of which the desktop is associated.

9. A device comprising:

at least two displays;

a memory;

a processor in communication with the memory and each of the at least two displays, the processor operable to:

provide a composite display, wherein the composite display includes the at least two displays;

receive a gesture on the device, wherein the gesture reveals a previously opened desktop, wherein the desktop occupies a substantial portion of the composite display;

associate the desktop with the composite display; and

afrer associating the desktop with the composite display, create a logic data structure associated with the desktop that describes the position of the desktop as being on top of a window stack and displayed across the at least two displays.

10. The device as defined in claim 9, wherein at least one other window is altered by the desktop reveal.

11. The device as defined in claim 10, further comprising altering a second logic data structure associated with the altered window.

12. The device as defined in claim 9, wherein the desktop is associated with all of the at least two displays.

13. The device as defined in claim 12, wherein the logic data structure comprises one or more of:

a window identifier adapted to identify the desktop in relation to other windows in the window stack;

a window stack position identifier adapted to identify the top position in the window stack for the desktop; and

at least two display identifiers adapted to identify each of the at least two displays of the device which the desktop is associated.

14. The device as defined in claim 13, wherein the desktop opened at a bottom of a window stack and is moved from the bottom of the window stack to the top of the window stack.

15. The device as defined in claim 14, wherein the desktop covers at least one other window.

16. A method for presenting a display for a device, the method comprising:

providing a composite display that spans at least a portion of two or more touch sensitive displays;

displaying a first window in at least one portion of the composite display;

receiving a first gesture associated with revealing a desktop;

interpreting the gesture as revealing a desktop; and

presenting the desktop across the composite display spanning at least the portion of the two or more touch sensitive displays, wherein the desktop obscures the first window.

17. The method defined in claim 16, further comprising:

receiving a second gesture;

interpreting the second gesture as moving the obscured window; and

presenting the obscured window as obscuring at least a portion of the reveal desktop.

18. The method defined in claim 17, wherein presenting the obscured window appears to cover at least a portion of the desktop.

19. The method defined in claim 18, wherein the first gesture is a selection in a configurable area, and wherein the desktop appears to expand over the first window.

20. The method defined in claim 18, wherein the second gesture is a drag gesture in a gesture capture region, and wherein the obscured window appears to slide over the revealed desktop.

ABSTRACT

Systems and methods are provides for revealing a desktop in a window stack for a multi-screen device. The window stack can change based on the revealing of a desktop. The system can receive a gesture indicating an application with the desktop, which was previously created in the stack, is to be revealed on the display of the device. Upon receiving the gesture, the system determines that the desktop is to occupy substantially all of a composite display that spans substantially all of the two or more touch sensitive displays of the device. Then, the system can determine that the desktop is to be associated with the composite display and change a logic data structure associated with the desktop to describe the position of the desktop on the top of the window stack.

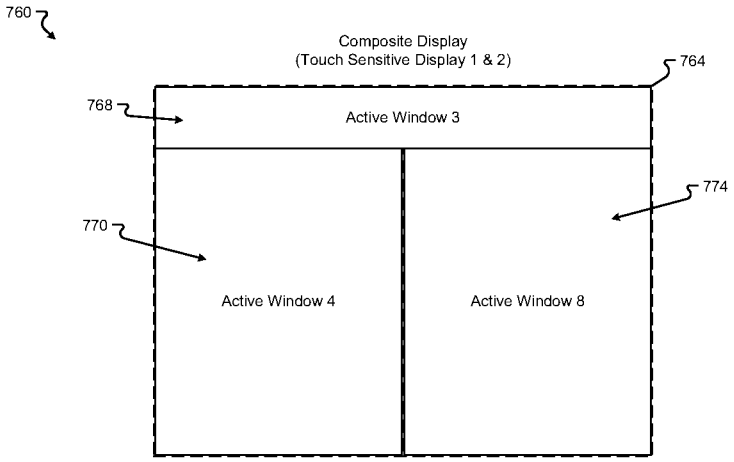


FIG. 7C

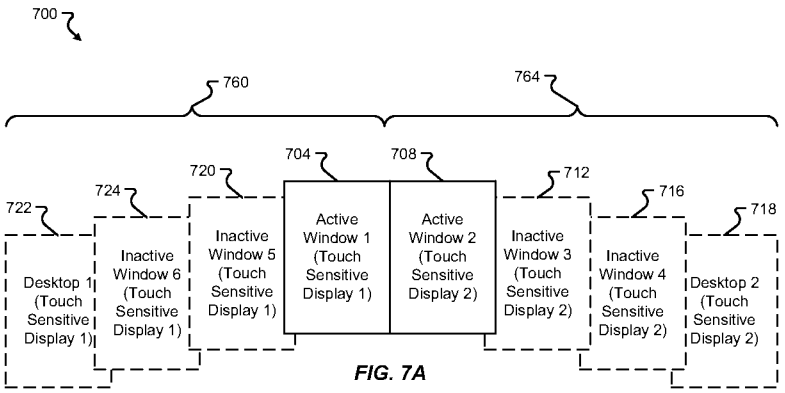


FIG. 7A

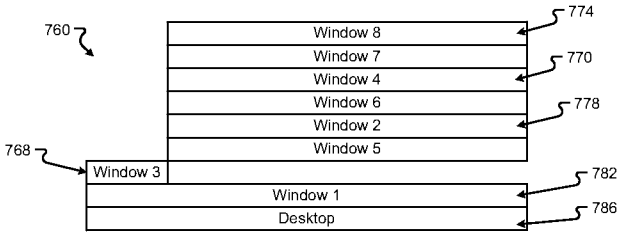


FIG. 7D

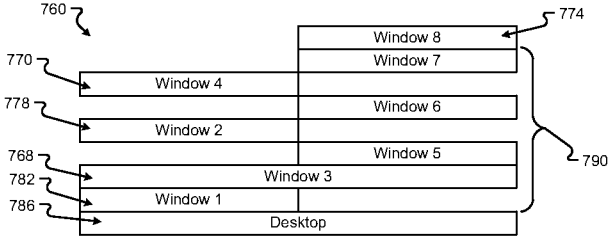


FIG. 7E

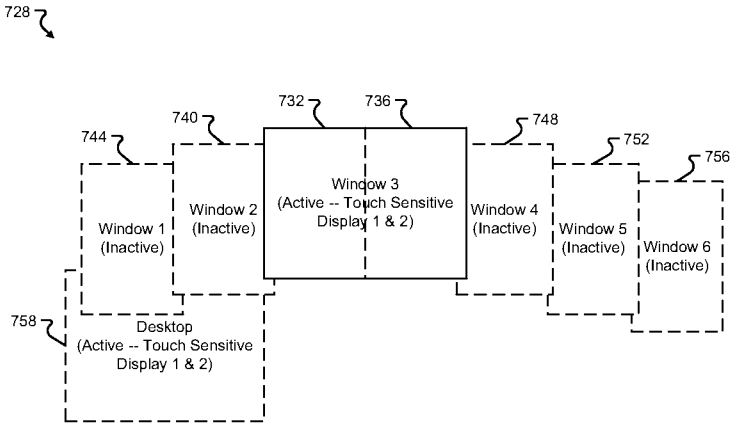


FIG. 7B

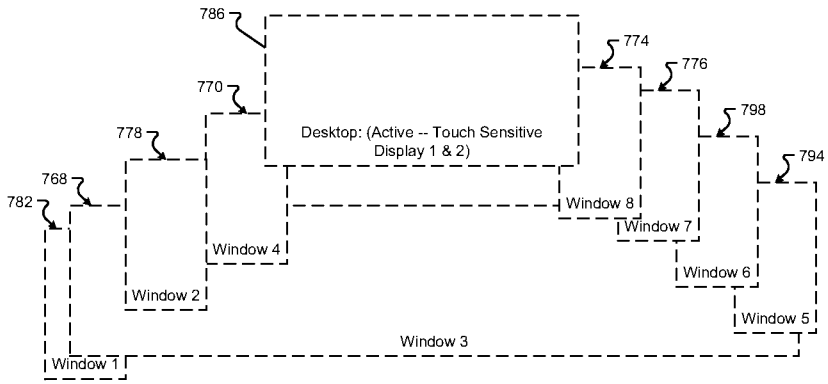


FIG. 7I

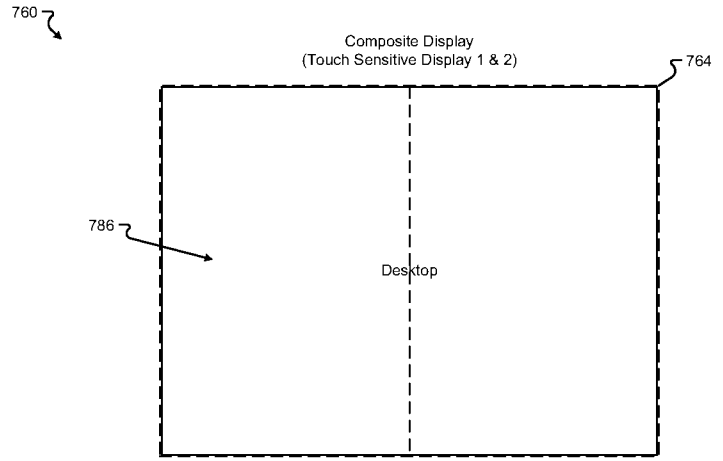


FIG. 7F

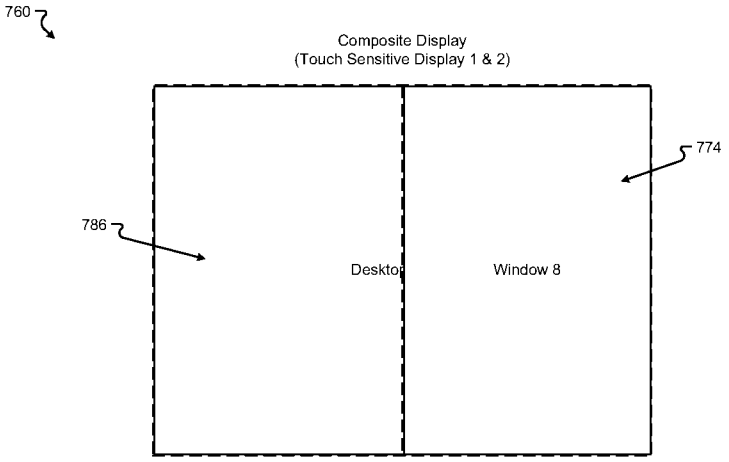


FIG. 7J

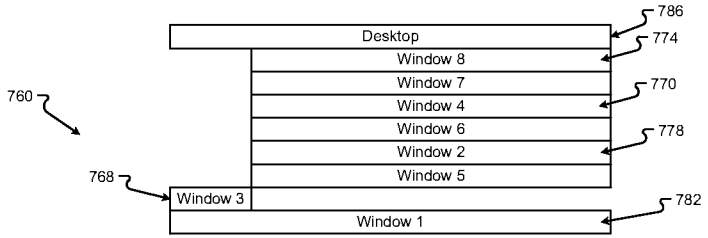


FIG. 7G

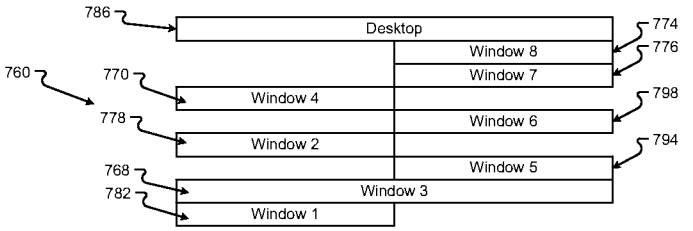


FIG. 7H

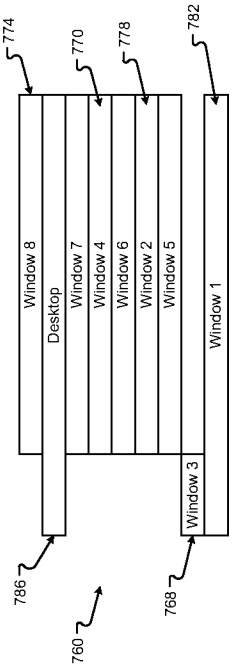


FIG. 7K

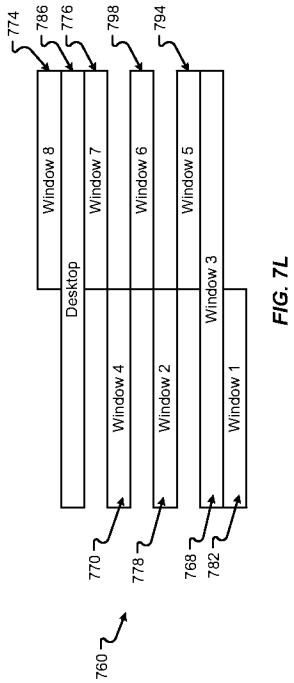


FIG. 7L

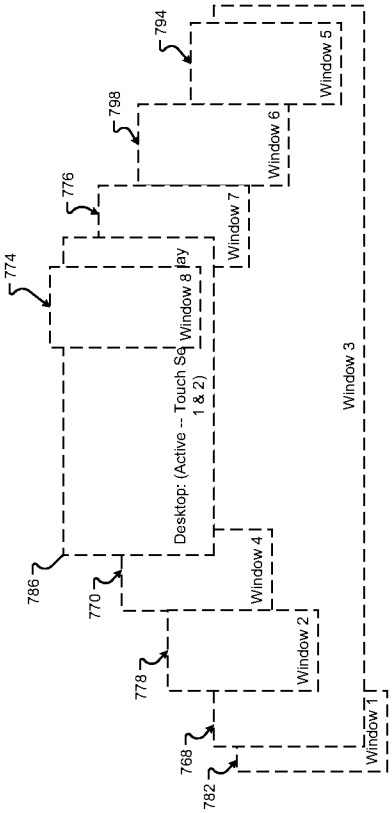


FIG. 7M

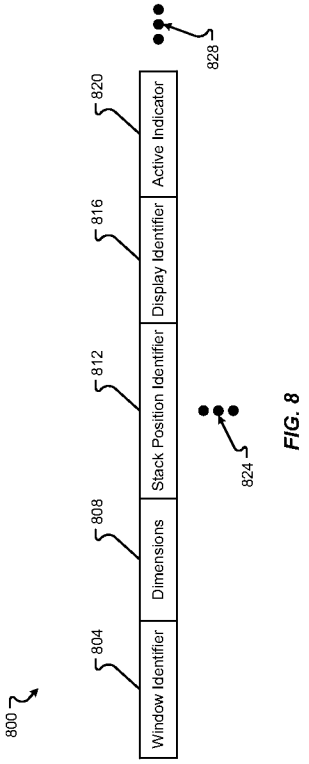


FIG. 8

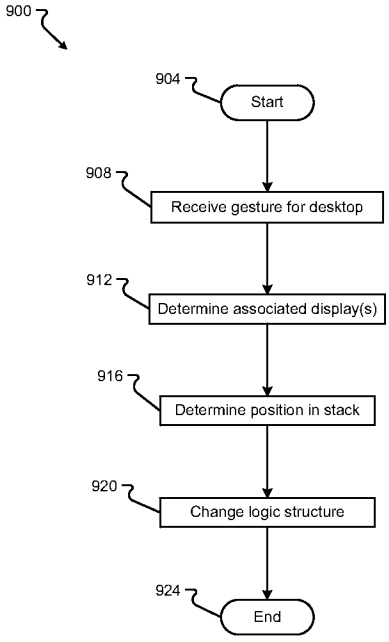


FIG. 9

DESKTOP REVEAL EXPANSION

[0100] Display Controls:

[0101] FIGS. 7A-C depict graphical representations of embodiments of a device 100 in a closed portrait state 304 where the primary screen 104 in a first portrait configuration 604 is shown surrounded by a solid line, while the secondary display 108 is in a second portrait configuration 608 but is not visible (also known as non-visible), and is accordingly represented by a surrounding dashed line. Other pages 704, 708, 712, 716 that are not visible on the primary screen 104 or the secondary screen 108 are also graphically represented as surrounded by dashed lines.

[0102] In one embodiment of the device 100, the MDM class 524 arranges applications, desktops, and/or other displayable information through the Window Management class 532 onto separate pages 704, 708, 712, 716 and organizes those pages 704, 708, 712, 716 into virtual "stacks." These stacks of pages can be created, deleted, shuffled, and/or moved by a user or an output from the MDM class 524, similar to a user manipulating the virtual playing cards of a deck stack. In FIGS. 7A-C, the stacks of individual pages 704, 708, 712, 716 are shown spread outwardly in a linear format on both the left side of the primary screen 104 as well the right side of the non-visible secondary screen 108 to better visualize the organization of the virtual stacks. The position of a page from the primary screen 104 or secondary screen 108 is the position of the page in each respective stack stored on the device 100. Therefore, if a page is graphically represented immediately adjacent to the primary screen 104, or immediately adjacent to the secondary screen 108, it is the first page in the stack that may be moved onto that adjacent screen with a user input gesture. Although only several pages of a stack are shown in FIGS. 7A-C, the number of pages available to be organized in the stack can vary and, in some embodiments, may equal the number of applications, desktops, and/or other displayable information created on the device 100. Moreover, the stack adjacent to the primary screen 104 and the stack adjacent to the secondary screen 108 may act as a unified stack (where a move or manipulation of a page from one stack moves or manipulates the pages in both stacks), or as independent stacks (each screen having its own independently navigable stack of pages).

[0103] In an embodiment of the present disclosure, it is anticipated that the concept of stacks disclosed herein may also be applied to desktops available on the device 100. In other words, a desktop virtual stack may be used for the desktops which can be displayed by the device 100 and

an application virtual stack may be used for the applications which can be displayed by the device 100. Specifically, when a desktop is revealed it may be divided into multiple pages for display on multiple screens of the device 100. These pages may be manipulated in their own desktop virtual stack, or stacks, separate from the application virtual stacks and/or other virtual stacks. This separation of stacks allows for greater user flexibility in navigating through different applications or desktops by creating an intuitive interface dependent on the data displayed to the screens 104, 108. However, it is also anticipated that these stacks could be combined to form a single virtual stack.

[0104] FIG. 7A shows a graphical representation of the device 100 running four separate applications. The device 100 is shown in a closed portrait state 304. In the present embodiment, only the primary screen 104 is visible showing a running application. Three pages are not visible but are all running different applications. Specifically, the non-visible secondary screen 108, a first stack first page 704, and a second stack first page 712, are running different applications (e.g., second application, third application, and fourth application).

[0105] FIG. 7B shows an embodiment of the present device 100 where the desktop reveal expansion has been initiated. The device 100 is shown in a closed portrait state 304. The desktop reveal expansion may be initiated by, (1) a user input gesture, (2) combination of user input gestures, (3) a memory output, (4) a response to a predetermined condition (e.g., application control, power levels, communications interrupt, operating system state, device screen state open/closed, timers, and single or multiple sensor outputs) or (5) any combination thereof, that the MDM class 524 registers and interprets as an initiation command. As illustrated in FIG. 7B, once the desktop reveal expansion is initiated, the MDM class 524 and the Window Management class 532 move the application running on the primary screen 104 to the first stack first page 704 (shown on the lefr-hand side of the primary screen 104) and the second application running on the secondary screen 108 to the second stack first page 712 (shown on the right-hand side of the secondary screen 108). At this time, the applications that were running on the first stack first page 704 and the second stack first page 712, are moved to a first stack second page 708 and a second stack second page 716, respectively. The first page of the desktop is now visible on the primary screen 104, while the non-visible secondary screen 108 is running the second page of the desktop. When the device 100 is closed, the MDM class 524 interprets the display state of the secondary screen 108, and if the secondary screen 108 is in an "off" state

(where the screen is not actively displaying information), the MDM class 524 will create a virtual representation of the display in its register through the Window Management class 532.

[0106] In embodiments where multiple pages are open in a stack, an initiation of the desktop reveal expansion will move each page one position further away from the primary and secondary screens 104, 108. It is anticipated that at least one embodiment of the present disclosure provides that pages in the first stack, as well as pages in the second stack, may move together as if the pages were virtually connected. Therefore, a single position move of one page in a stack correspondingly moves all of the pages in that stack by one position. In addition, it is anticipated that this concept can also be applied to the pages of a virtual desktop stack.

[0107] FIG. 7C shows an embodiment of the present device 100, in a closed portrait state 304, where the desktop reveal expansion has been initiated and the desktop has been indexed to display the second desktop page on the primary screen 104. As the second desktop is indexed onto the primary screen 104, a third desktop is indexed onto the non-visible secondary screen 108 by the MDM class 524 and the Window Management class 532. While the desktop reveal expansion is initiated, the user can navigate between desktop pages by moving them into view on the primary screen 104 by providing a user input gesture 720. This input may be detected by one or more of the hardware components of the device 100 as discussed herein.

[0108] FIG. 8A shows a graphical representation of the device 100 running four separate single-screen applications in an open portrait state 320. In the present embodiment, both the primary screen 104 and secondary screen 108 are visible showing running applications. Two pages are not visible but are running different applications. Specifically, the first stack first page 704, and the second stack first page 712, are running different applications (e.g., second application and third application).

[0109] FIG. 8B shows an embodiment of the present device 100 where the desktop reveal expansion has been initiated. The device 100 is shown in an open portrait state 320. The desktop reveal expansion may be initiated by, (1) a user input gesture, (2) combination of user input gestures, (3) a memory output, (4) a response to a predetermined condition (e.g., application control, power levels, communications interrupt, operating system state, device screen state open/closed, timers, and single or multiple sensor outputs) or (5) any combination thereof, that the MDM class 524 registers and interprets as an initiation command. As illustrated in FIG. 8B, once the desktop reveal expansion is initiated, the MDM class 524 and the Window

Management class 532 move the application running on the primary screen 104 to the first stack first page 704 (shown on the lefr-hand side of the primary screen) and the second application running on the secondary screen 108 to the second stack first page 712 (shown on the right-hand side of the secondary screen). At this time, the applications that were running on the first stack first page 704 and the second stack first page 712, are moved to a first stack second page 708 and a second stack second page 716, respectively. The first page of the desktop is now visible on the primary screen 104, while the second page of the desktop is visible on the secondary screen 108. It is anticipated that at least one embodiment of the present disclosure provides that pages in the lefr stack, as well as pages in the right stack, may move together as if the pages in a stack are virtually connected. Therefore, a single position move of a page in one stack correspondingly moves all of the pages in that stack by one position.

[0110] FIG. 8C shows an embodiment of the present device 100, in an open portrait state 320, where the desktop reveal expansion has been initiated and the desktop has been indexed to display the second desktop page on the primary screen 104 and the third desktop on the secondary screen 108. While the desktop reveal expansion is initiated, the user can navigate between desktop pages by moving them into view on the primary screen 104 and/or secondary screen 108 by providing a user input gesture 720. This input may be detected by one or more of the hardware components of the device 100 as discussed herein. In one embodiment of the present invention, the desktops may move together when being navigated by the user. This linked movement of desktops is similar to embodiments in which the pages in a stack can move together as disclosed herein.

[0111] Referring now to Fig. 9, a desktop reveal expansion method 900 will be described in accordance with at least some embodiments of the present disclosure. The method is initiated at step 904. Specifically, a device 100 may be initially running applications on one or more screens 104, 108 in an open or closed portrait state 304, 320. The method continues when the device 100 detects a desktop reveal input (step 908) from the user. In some embodiments, this input is registered at the MDM class 524.

[0112] Next, the MDM class 524 must detect any applications open or running on the primary screen 104, the secondary screen 108, the first stack, and the second stack. Once detected, the MDM class 524 and Window Management class 532 moves all of the application pages one position further away from their initial positions adjacent to the primary and secondary screens

104, 108 (step 916). Specifically, the first stack is moved to the left of the primary screen 104 and the second stack is moved to the right of the secondary screen 108.

[0113] After or while the Window Management class 532 moves the application pages, the method continues with the MDM class 524 and Window Management class 532 displaying the first desktop on the primary screen 104 and the second desktop on the secondary screen 108 (step 920). If the device 100 is closed, and at least one display is turned off, then the desktop will only be displayed on the screen that is on.

[0114] Once the desktop is revealed, the MDM class 524 detects a user gesture input (step 924). The nature of the user gesture input will determine whether to display the next desktop in order (step 928), indicate the end of the available desktop range (step 932), or deactivate the desktop reveal and return the previously open applications to their original pages (step 936) which were detected earlier in the process (at step 912).

[0115] Referring to Fig. 10, a desktop reveal expansion method 1000 will be described in accordance with at least some embodiments of the present disclosure. The method is initiated at step 1004. This method is initiated if the device 100 is already running with the desktop revealed. The method continues when the device 100 detects a user gesture input (step 1008). The MDM class 524 determines and then registers the input provided by a user, which allows the user to navigate through multiple desktop pages or cancel the desktop reveal expansion (step 936).

[0116] After or while the MDM class 524 determines the desktop number displayed on the primary screen 104, the MDM class 524 must determine if the displayed desktop is the first, last, or one of the desktops between the first and last desktops (steps 1016, 1028, 1040). If the first desktop is displayed 1016, the MDM class 524 interprets the previously registered gesture input and either displays the next desktop in order 1020, or indicates to the user that the desktop is at its end of range 1024. If the last desktop is displayed 1028, the MDM class 524 interprets the previously registered gesture input and either indicates to the user that the desktop is at its end of range 1032, or it displays the previous desktop in order 1036. If the desktop displayed on the primary screen 104 is neither the first or last desktop, then the MDM class 524 interprets the previously registered gesture input to either display the next desktop in order 1044 or display the previous desktop in order 1048.

[0117] After the MDM class 524 interprets the previously registered gesture input and displays the appropriate response, the MDM class 524 returns to determine user gesture input (step 1008) until the method is ended (step 936).

[0118] The exemplary systems and methods of this disclosure have been described in relation to display controls for a multi-screen device. However, to avoid unnecessarily obscuring the present disclosure, the preceding description omits a number of known structures and devices. This omission is not to be construed as a limitation of the scopes of the claims. Specific details are set forth to provide an understanding of the present disclosure. It should however be appreciated that the present disclosure may be practiced in a variety of ways beyond the specific detail set forth herein.

[0119] Furthermore, while the exemplary aspects, embodiments, and/or configurations illustrated herein show the various components of the system collocated, certain components of the system can be located remotely, at distant portions of a distributed network, such as a LAN and/or the Internet, or within a dedicated system. Thus, it should be appreciated, that the components of the system can be combined in to one or more devices, such as a Personal Computer (PC), laptop, netbook, Personal Digital Assistant (PDA), tablet, etc., or collocated on a particular node of a distributed network, such as an analog and/or digital telecommunications network, a packet-switch network, or a circuit-switched network. It will be appreciated from the preceding description, and for reasons of computational efficiency, that the components of the system can be arranged at any location within a distributed network of components without affecting the operation of the system. For example, the various components can be located in a switch such as a PBX and media server, gateway, in one or more communications devices, at one or more users' premises, or some combination thereof. Similarly, one or more functional portions of the system could be distributed between a telecommunications device(s) and an associated computing device.

[0120] Furthermore, it should be appreciated that the various links connecting the elements can be wired or wireless links, or any combination thereof, or any other known or later developed element(s) that is capable of supplying and/or communicating data to and from the connected elements. These wired or wireless links can also be secure links and may be capable of communicating encrypted information. Transmission media used as links, for example, can be any suitable carrier for electrical signals, including coaxial cables, copper wire and fiber optics,

and may take the form of acoustic or light waves, such as those generated during radio-wave and infra-red data communications.

[0121] Also, while the flowcharts have been discussed and illustrated in relation to a particular sequence of events, it should be appreciated that changes, additions, and omissions to this sequence can occur without materially affecting the operation of the disclosed embodiments, configuration, and aspects.

[0122] A number of variations and modifications of the disclosure can be used. It would be possible to provide for some features of the disclosure without providing others.

[0123] In some embodiments, the systems and methods of this disclosure can be implemented in conjunction with a special purpose computer, a programmed microprocessor or microcontroller and peripheral integrated circuit element(s), an ASIC or other integrated circuit, a digital signal processor, a hard-wired electronic or logic circuit such as discrete element circuit, a programmable logic device or gate array such as PLD, PLA, FPGA, PAL, special purpose computer, any comparable means, or the like. In general, any device(s) or means capable of implementing the methodology illustrated herein can be used to implement the various aspects of this disclosure. Exemplary hardware that can be used for the disclosed embodiments, configurations and aspects includes computers, handheld devices, telephones (e.g., cellular, Internet enabled, digital, analog, hybrids, and others), and other hardware known in the art. Some of these devices include processors (e.g., a single or multiple microprocessors), memory, nonvolatile storage, input devices, and output devices. Furthermore, alternative software implementations including, but not limited to, distributed processing or component/object distributed processing, parallel processing, or virtual machine processing can also be constructed to implement the methods described herein.

[0124] In yet another embodiment, the disclosed methods may be readily implemented in conjunction with software using object or object-oriented software development environments that provide portable source code that can be used on a variety of computer or workstation platforms. Alternatively, the disclosed system may be implemented partially or fully in hardware using standard logic circuits or VLSI design. Whether software or hardware is used to implement the systems in accordance with this disclosure is dependent on the speed and/or efficiency requirements of the system, the particular function, and the particular software or hardware systems or microprocessor or microcomputer systems being utilized.

[0125] In yet another embodiment, the disclosed methods may be partially implemented in software that can be stored on a storage medium, executed on programmed general-purpose computer with the cooperation of a controller and memory, a special purpose computer, a microprocessor, or the like. In these instances, the systems and methods of this disclosure can be implemented as program embedded on personal computer such as an applet, JAVA® or CGI script, as a resource residing on a server or computer workstation, as a routine embedded in a dedicated measurement system, system component, or the like. The system can also be implemented by physically incorporating the system and/or method into a software and/or hardware system.

[0126] Although the present disclosure describes components and functions implemented in the aspects, embodiments, and/or configurations with reference to particular standards and protocols, the aspects, embodiments, and/or configurations are not limited to such standards and protocols. Other similar standards and protocols not mentioned herein are in existence and are considered to be included in the present disclosure. Moreover, the standards and protocols mentioned herein and other similar standards and protocols not mentioned herein are periodically superseded by faster or more effective equivalents having essentially the same functions. Such replacement standards and protocols having the same functions are considered equivalents included in the present disclosure.

[0127] The present disclosure, in various aspects, embodiments, and/or configurations, includes components, methods, processes, systems and/or apparatus substantially as depicted and described herein, including various aspects, embodiments, configurations embodiments, subcombinations, and/or subsets thereof. Those of skill in the art will understand how to make and use the disclosed aspects, embodiments, and/or configurations after understanding the present disclosure. The present disclosure, in various aspects, embodiments, and/or configurations, includes providing devices and processes in the absence of items not depicted and/or described herein or in various aspects, embodiments, and/or configurations hereof, including in the absence of such items as may have been used in previous devices or processes, e.g., for improving performance, achieving ease and\or reducing cost of implementation.

[0128] The foregoing discussion has been presented for purposes of illustration and description. The foregoing is not intended to limit the disclosure to the form or forms disclosed herein. In the foregoing Detailed Description for example, various features of the disclosure are

grouped together in one or more aspects, embodiments, and/or configurations for the purpose of streamlining the disclosure. The features of the aspects, embodiments, and/or configurations of the disclosure may be combined in alternate aspects, embodiments, and/or configurations other than those discussed above. This method of disclosure is not to be interpreted as reflecting an intention that the claims require more features than are expressly recited in each claim. Rather, as the following claims reflect, inventive aspects lie in less than all features of a single foregoing disclosed aspect, embodiment, and/or configuration. Thus, the following claims are hereby incorporated into this Detailed Description, with each claim standing on its own as a separate preferred embodiment of the disclosure.

[0129] Moreover, though the description has included description of one or more aspects, embodiments, and/or configurations and certain variations and modifications, other variations, combinations, and modifications are within the scope of the disclosure, e.g., as may be within the skill and knowledge of those in the art, after understanding the present disclosure. It is intended to obtain rights which include alternative aspects, embodiments, and/or configurations to the extent permitted, including alternate, interchangeable and/or equivalent structures, functions, ranges or steps to those claimed, whether or not such alternate, interchangeable and/or equivalent structures, functions, ranges or steps are disclosed herein, and without intending to publicly dedicate any patentable subject matter.

What is claimed is:

1. A method of displaying a desktop on a multi-screen device, comprising:

receiving a first predetermined input that represents an instruction to reveal a desktop on a first screen of the multi-screen device;

determining a first desktop to display on the first screen;

determining a second desktop to display on a second screen of the multi-screen device; and

responding to the first predetermined input with an output that causes the first desktop to be displayed on the first screen and which also causes the second desktop to be at least one of (i) actually displayed on the second screen and (ii) virtually displayed on the second screen.

2. The method of claim 1, wherein the first predetermined input comprises at least one of the following: i) a user input gesture; ii) a combination of user input gestures; iii) a memory output; iv) a response to a programmed condition; and v) hardware timers.

3. The method of claim 1, wherein responding to the first predetermined input comprises at least one of the following: i) referring to a memory; ii) selecting an output; and iii) providing the output as an electrical signal.

4. The method of claim 1, wherein the second desktop is virtually displayed by:

registering in memory the second desktop for display by the second screen;

preventing the second desktop from displaying on the second screen where the second screen is determined to be inactive; and

wherein access to the memory is provided such that the second desktop can be retrieved.

5. The method of claim 1, wherein the second desktop is actually displayed by:

registering in memory the second desktop for display by the second screen where the second screen is determined to be active; and

displaying the second desktop on the second screen.

6. The method of claim 1, further comprising:

receiving a second predetermined input afrer the first predetermined input, the second predetermined input representing an instruction to navigate a plurality of desktops;

discontinuing display of the first desktop on the first screen and storing the first desktop in a virtual desktop stack memory;

replacing the first desktop on the first screen with the second desktop from the second screen for display on the first screen;

displaying a third desktop to be at least one of (i) actually displayed on the second screen and (ii) virtually displayed on the second screen.

7. The method of claim 6, wherein the third desktop is virtually displayed on the second screen while the multi-screen device is in a closed state.

8. The method of claim 6, wherein the third desktop is actually displayed on the second screen while the multi-screen device is in an open state.

9. The method of claim 1, wherein display rules are referenced and followed according to:

a) when both screens are active the first desktop is actually displayed on the first screen and the second desktop is actually displayed on the second screen; and

b) when the second screen is inactive the first desktop is actually displayed on the first screen and the second desktop is virtually displayed on the second screen.

10. A non-transitory computer-readable medium having stored thereon instructions that cause a computing system to execute a method, the instructions comprising:

instructions configured to receive a first predetermined input that represents an instruction to reveal a desktop on a first screen of the multi-screen device;

instructions configured to determine a first desktop to display on the first screen;

instructions configured to determine a second desktop to display on a second screen of the multi-screen device; and

instructions configured to respond to the first predetermined input with an output that causes the first desktop to be displayed on the first screen and which also causes the second desktop to be at least one of (i) actually displayed on the second screen and (ii) virtually displayed on the second screen.

11. The computer-readable medium of claim 10, wherein the first screen corresponds to a primary screen and wherein the second screen corresponds to a secondary screen.

12. The computer-readable medium of claim 10, further comprising:

instructions configured to receive a second predetermined input that represents an instruction to move the second desktop from the second screen to be displayed on the first screen of the multi-screen device.

13. The computer-readable medium of claim 10, further comprising:

instructions configured to determine whether the multi-screen device is in an open or closed state;

instructions configured to actually display the first desktop on the first screen and actually display the second desktop on the second screen when the multi-screen device is in the open state; and

instructions configured to actually display the first desktop on the first screen and virtually display the second desktop on the second screen when the multi-screen device is in the closed state.

14. The computer-readable medium of claim 10, further comprising:

instructions configured to determine whether the second screen is active or inactive;

instructions configured to actually display the first desktop on the first screen and actually display the second desktop on the second screen when both screens are active; and

instructions configured to actually display the first desktop on the first screen and virtually display the second desktop on the second screen when the first screen is active and the second screen is inactive.

15. The computer-readable medium of claim 14, further comprising:

instructions configured to receive a second predetermined input after the first predetermined input, the second predetermined input representing an instruction to navigate a plurality of desktops;

instructions configured to move the first desktop off the first screen and store the first desktop in a virtual desktop stack memory;

instructions configured to replace the first desktop on the first screen with the second desktop from the second screen for display on the first screen; and

instructions configured to introduce a third desktop to be at least one of (i) actually displayed on the second screen and (ii) virtually displayed on the second screen.

16. A multi-screen user device, comprising:

a first screen including a first display area;

a second screen including a second display area;

a hinge configured to connect the first screen with the second screen;

a first sensor configured to produce a first signal indicative of a position of the hinge;

a second sensor configured to produce a second signal indicative of a relative position of the first screen and the second screen; and

a computer-readable medium having instructions stored thereon that include:

a first set of instructions configured to determine, based at least partially on the first signal, whether the hinge is in a first position or second position; and

a second set of instructions configured to determine, based on the first signal, whether to actually display or virtually display a second desktop on the second screen.

17. The device of claim 16, wherein the first screen corresponds to a primary screen and wherein the second screen corresponds to a secondary screen.

18. The device of claim 16, further comprising:

a first user input gesture area of the first screen;

a second user input gesture area of the second screen; and

wherein the first and second user input gesture areas are configured to accept input from a user.

19. The device of claim 18, wherein the first user input gesture area is located on the first screen and outside of the first display area.

20. The device of claim 18, wherein the second user input gesture area is located on the second screen and outside of the second display area.

ABSTRACT

A dual-screen user device and methods for revealing a combination of desktops on single and multiple screens are disclosed. Specifically, a determined number of desktops is displayed to at least one of the screens of the device conditioned upon input received and the state of the device. Where a screen of the device is determined to be inactive, the desktop is not displayed to the screen, but is stored in a virtually displayed state by the device. Upon receiving input that the inactive screen is active, the device can actually display the desktop to the screen.

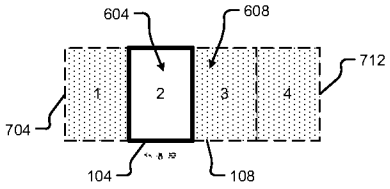


FIG. 7A

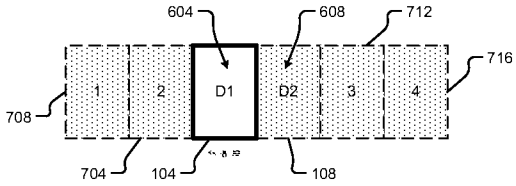


FIG. 7B

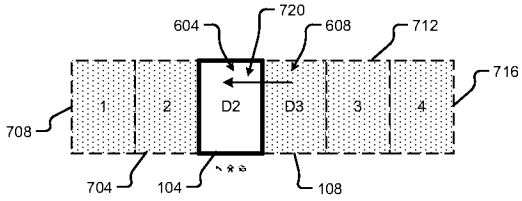


FIG. 7C

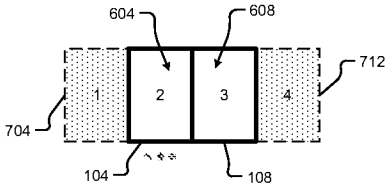


FIG. 8A

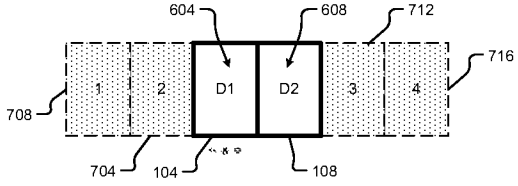


FIG. 8B

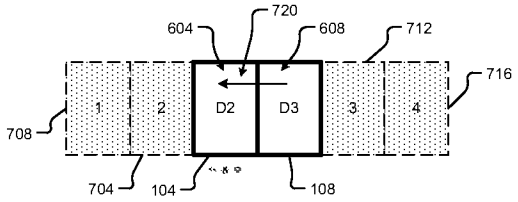


FIG. 8C

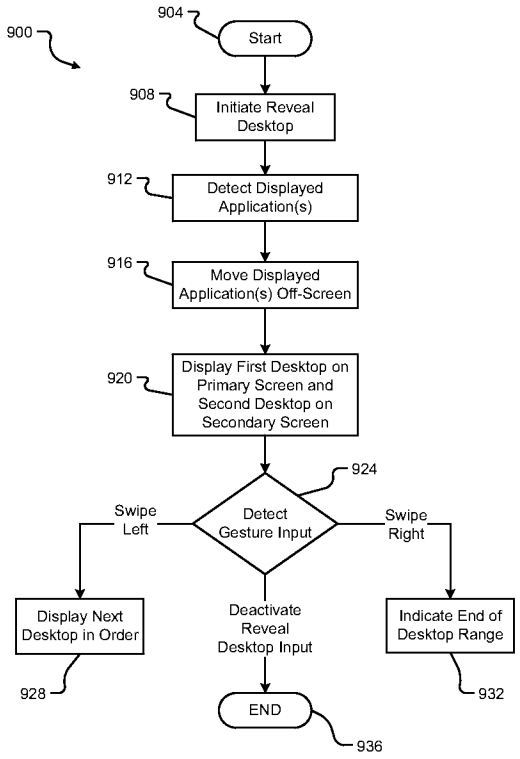


FIG. 9

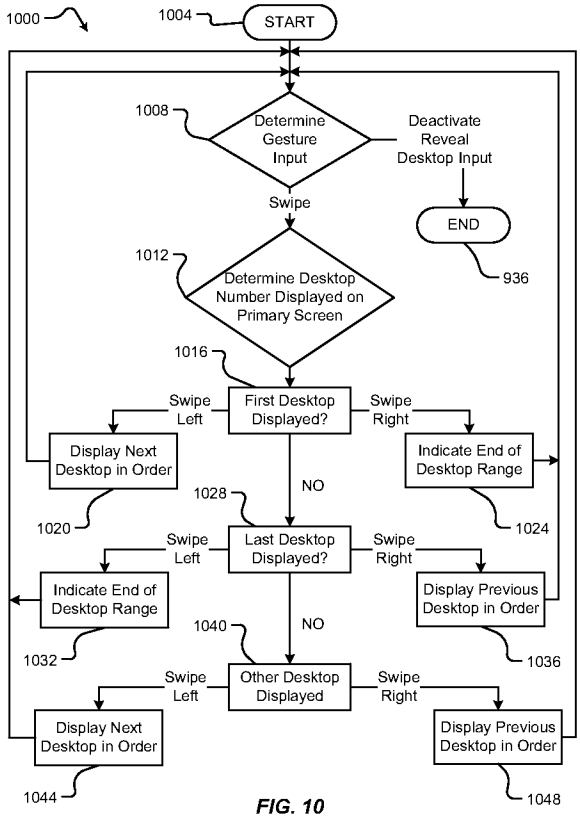


FIG. 10

LAUNCHING APPLICATIONS INTO REVEALED DESKTOP

[0100] Display Controls:

[0001] FIGS. 7A-C depict graphical representations of embodiments of a device 100 in an open portrait state 320 where the primary screen 104 displays application data in a first portrait configuration 604 and the secondary screen 108 displays application data in a second portrait configuration 608. Other pages 704, 712 that are not visible on the primary screen 104 or the secondary screen 108 are graphically represented as surrounded by dashed lines.

[0101] In one embodiment of the device 100, the MDM class 524 arranges applications, desktops, and/or other displayable information through the Window Management class 532 onto separate pages 704, 708, 712, 716 and organizes those pages 704, 708, 712, 716 into virtual "stacks." These stacks allow pages to be created, deleted, shuffled, and/or moved by a user or an output from the MDM class 524, similar to a user manipulating the virtual playing cards of a deck stack. In FIGS. 7A-C, the stacks of individual pages 704, 708, 712, 716 are shown spread outwardly in a linear format on both the left side of the primary screen 104 as well the right side of the secondary screen 108 to better visualize the organization of the virtual stacks. The position of a page from the primary screen 104 or secondary screen 108 is the position of the page in each respective stack stored on the device 100. Therefore, if a page is graphically represented immediately adjacent to the primary screen 104, or immediately adjacent to the secondary screen 108, it is the first page in the stack that may be moved onto that adjacent screen with a user input gesture. Although only several pages of a stack are shown in FIGS. 7A-C, the number of pages available to be organized in the stack can vary and, in some embodiments, may equal the number of applications, desktops, and/or other displayable information created on the device 100. Moreover, the stack adjacent to the primary screen 104 and the stack adjacent to the secondary screen 108 may act as a unified stack (where a move or manipulation of a page from one stack moves or manipulates the pages in both stacks), or as independent stacks (each screen having its own independently navigable stack of pages).

[0102] In an embodiment of the present disclosure, it is anticipated that the concept of stacks disclosed herein may also be applied to desktops available on the device 100. In other words, a desktop virtual stack may be used for the desktops which can be displayed by the device 100 and an application virtual stack may be used for the applications which can be displayed by the device 100. Specifically, when a desktop is revealed it may be divided into multiple pages for

display on multiple screens of the device 100. These pages may be manipulated and/or stored in their own desktop virtual stacks separate from the application virtual stacks and/or other virtual stacks. This separation of stacks allows for greater user flexibility in navigating through different applications or desktops by creating an intuitive interface dependent on the data displayed to the screens 104, 108. However, it is also anticipated that these stacks could be combined to form a single virtual stack.

[0103] FIG. 7A shows a graphical representation of the device 100 running three separate applications. As shown, the first application is running in a dual-screen mode, also known as maximized, on both the primary screen 104 and the secondary screen 108. The device 100 is shown in an open portrait state 320. In the present embodiment, both the primary screen 104 and secondary screen 108 are visible while the first application is running. Although not displayed to the primary screen 104 or secondary screen 108, two applications are running on separate pages in the virtual stack. Specifically, the first stack first page 704 and the second stack first page 712 are running different applications (e.g., second application and third application).

[0104] FIG. 7B shows an embodiment of the present device 100 where the desktop reveal expansion has been initiated. The device 100 is shown in an open portrait state 320. The desktop reveal expansion may be initiated by, (1) a user input gesture, (2) combination of user input gestures, (3) a memory output, (4) a response to a predetermined condition (e.g., application control, power levels, communications interrupt, operating system state, device screen state open/closed, timers, and single or multiple sensor outputs) or (5) any combination thereof, that the MDM class 524 registers and interprets as an initiation command. As illustrated in FIG. 7B, once the desktop reveal expansion is initiated, the MDM class 524 and the Window Management class 532 split the dual-screen application into two separate pages and move the first page of the dual-screen application to the first stack first page 704 (shown on the lefr-hand side of the screen) and the second page of the dual-screen application to the second stack first page 712 (shown on the right-hand side of the screen). At this time, the applications that were running on the first stack first page 704 and the second stack first page 712, are moved to the first stack second page 708 and the second stack second page 716, respectively. The first and second page of the desktop is now visible on the primary screen 104 and the secondary screen 108, respectively. From either screen 104, 108 of the displayed desktop, a user may select an application by providing a user input gesture 720. This user input gesture 720 is registered by

the MDM class 524 where it is interpreted as a command to deactivate the reveal desktop and open a new application on the screen from which the user input gesture 720 was detected. In some embodiments, the new application may display onto a screen other than that from which the user input was detected. Specifically, it is anticipated that the MDM class 524 can determine through logic, rules, user input, or combinations thereof to open a new application on either screen of the device 100, in accordance with predetermined conditions.

[0105] In embodiments where multiple pages are open in the first and second stack, an initiation of the desktop reveal expansion will move each page one position further away from the primary and secondary screens 104, 108, respectively. It is anticipated that at least one embodiment of the present disclosure provides that pages in the first stack, as well as pages in the second stack, may move together as if the pages were virtually connected. Therefore, a single position move of one page in a stack correspondingly moves all of the pages in that stack and the other stack by a single position. In addition, it is anticipated that this concept can also be applied to the pages of a virtual desktop stack.

[0106] FIG. 7C shows an embodiment of the present device 100, in an open portrait state 320, wherein a new application has been chosen from the secondary screen 108 and is transitioning 724 onto the secondary screen 108 where it will be displayed. As the new application is transitioning 724 onto the secondary screen 108, the reveal desktop is deactivated by the MDM class 524 and the dual-screen application is displayed on the primary screen 104 by the Window Management class 532. In at least some embodiments, the visual content of the dual-screen application presented by the MDM class 524 and the Window Management class 532 may depend on the screen from which the new application is chosen by the user input gesture 720. In some cases, the MDM class 524 and Window Management class 532 may present the first stack information or the second stack information as defined by protocol and/or logic.

[0107] Referring now to FIG. 8, a method for launching applications into reveal desktop 800 will be described in accordance with at least some embodiments of the present disclosure. The method is initiated at step 804. Specifically, a device 100 is depicted as initially running a dual-screen application spread across the primary screen 104 and the secondary screen 108 while in an open portrait state 320. The method continues when the device 100 detects a desktop reveal input (step 812) from the user. In some embodiments, this input is registered at the MDM class 524.

[0108] Next, the MDM class 524 must move the dual-screen application off the primary and secondary screens 104, 108 onto a virtual stack or stacks. In some embodiments, the dual-screen application may be logically divided into two separate pages where one of the pages is stored in the first stack and the other page is stored in the second stack. As the dual-screen application is removed from display (step 816 and 820), the desktop is revealed on the primary and secondary screens 104, 108 (step 824).

[0109] Once the desktop is revealed, the MDM class 524 detects a user gesture input (step 828). The nature of the user gesture input will determine whether the reveal desktop function is deactivated or if an application is chosen to launch 836 from the desktop. In the event that a user selects to launch an application from the desktop, the MDM class 524 must make at least two determinations. First, which application was chosen by the user and second, from which screen was the user gesture input made. Once the MDM class 524 registers the application selection and user gesture input location, the MDM class 524 and the Window Management class 532, uses the information to launch the chosen application for display by the screen from which it was chosen. For example, if the user selected an application from the secondary screen 108, the application would launch and be displayed on the secondary screen 108. In contrast, if the user selected an application from the primary screen 104, then the application would launch and be displayed on the primary screen 104.

[0110] Afrer or while the MDM class 524 and Window Management class 532 launches and displays the application to the chosen screen, the reveal desktop is deactivated and one of the dual-screen applications (stored in the virtual stack) is moved onto the display that was not chosen. In some embodiments, the device 100 may be running and displaying a single-screen application on the primary screen 104 and a single-screen application on the secondary display 108. In which case, when the desktop is first revealed, the applications are moved from their respective screens into their respective virtual stacks. Next, and as described above, when a new application is launched from the revealed desktop it displays on the screen from which it was chosen. However, the other screen will display one of the previously running and displayed applications. The decision of which application to display on the other screen depends on the application priority, the sequence of applications created, and/or other determinative functions.

[0111] FIG. 9 shows a method for launching applications into reveal desktop 900 in accordance with at least some embodiments of the present disclosure. The method is initiated at

step 904. Specifically, a device 100 may be in the process of displaying application data to the primary screen 104 and/or the secondary screen 108. Any application data displayed to either the primary screen 104 and/or the secondary screen 108 is detected (step 908). The method continues when the device 100 detects a desktop reveal input (step 912) from the user. In some embodiments, this input is registered at the MDM class 524.

[0112] Next, if the device 100 was displaying an application to either screen 104, 108, as detected at step 908, the MDM class 524 must move the application off the primary and/or secondary screen 104, 108 onto a virtual stack or stacks. In some embodiments, an application displayed to the primary screen 104 may be stored in the first stack while an application displayed to the secondary screen 108 may be stored in the second stack. As an application is removed from display (step 920), the desktop is revealed on the primary and secondary screens 104, 108 (step 924). In the event that the device 100 is not displaying application data, the desktop is immediately revealed on the primary and secondary screens 104, 108 (step 924).

[0113] Once the desktop is revealed, the MDM class 524 detects a user gesture input (step 928). The nature of the user gesture input will determine whether the reveal desktop function is deactivated or if an application is chosen to launch 936 from the desktop. In the event that a user selects to launch an application from the desktop, the MDM class 524 must make at least two determinations. First, which application was chosen by the user and second, which screen will display the application. In some embodiments, the second determination may result in the application being displayed on a screen in accordance with certain predetermined conditions. These predetermined conditions may include rules, logic, application priority, applications creation sequence, and/or other determinative functions. Once the MDM class 524 registers the application selection in accordance with the predetermined conditions, the MDM class 524 and the Window Management class 532, uses the information to launch the chosen application for display by a specific screen (step 940).

[0114] After or while the MDM class 524 and Window Management class 532 launches and displays the application to the specific screen, the reveal desktop is deactivated and the other screen is configured to display a previously detected application (at step 908) or other data (step 948). The decision of which application, or data, to display on the other screen may depend on the application priority, the sequence of applications created, and/or other determinative functions.

[0115] The exemplary systems and methods of this disclosure have been described in relation to display controls for a multi-screen device. However, to avoid unnecessarily obscuring the present disclosure, the preceding description omits a number of known structures and devices. This omission is not to be construed as a limitation of the scopes of the claims. Specific details are set forth to provide an understanding of the present disclosure. It should however be appreciated that the present disclosure may be practiced in a variety of ways beyond the specific detail set forth herein.

[0116] Furthermore, while the exemplary aspects, embodiments, and/or configurations illustrated herein show the various components of the system collocated, certain components of the system can be located remotely, at distant portions of a distributed network, such as a LAN and/or the Internet, or within a dedicated system. Thus, it should be appreciated, that the components of the system can be combined in to one or more devices, such as a Personal Computer (PC), laptop, netbook, Personal Digital Assistant (PDA), tablet, etc., or collocated on a particular node of a distributed network, such as an analog and/or digital telecommunications network, a packet-switch network, or a circuit-switched network. It will be appreciated from the preceding description, and for reasons of computational efficiency, that the components of the system can be arranged at any location within a distributed network of components without affecting the operation of the system. For example, the various components can be located in a switch such as a PBX and media server, gateway, in one or more communications devices, at one or more users' premises, or some combination thereof. Similarly, one or more functional portions of the system could be distributed between a telecommunications device(s) and an associated computing device.

[0117] Furthermore, it should be appreciated that the various links connecting the elements can be wired or wireless links, or any combination thereof, or any other known or later developed element(s) that is capable of supplying and/or communicating data to and from the connected elements. These wired or wireless links can also be secure links and may be capable of communicating encrypted information. Transmission media used as links, for example, can be any suitable carrier for electrical signals, including coaxial cables, copper wire and fiber optics, and may take the form of acoustic or light waves, such as those generated during radio-wave and infra-red data communications.

[0118] Also, while the flowcharts have been discussed and illustrated in relation to a particular sequence of events, it should be appreciated that changes, additions, and omissions to this sequence can occur without materially affecting the operation of the disclosed embodiments, configuration, and aspects.

[0119] A number of variations and modifications of the disclosure can be used. It would be possible to provide for some features of the disclosure without providing others.

[0120] In some embodiments, the systems and methods of this disclosure can be implemented in conjunction with a special purpose computer, a programmed microprocessor or microcontroller and peripheral integrated circuit element(s), an ASIC or other integrated circuit, a digital signal processor, a hard-wired electronic or logic circuit such as discrete element circuit, a programmable logic device or gate array such as PLD, PLA, FPGA, PAL, special purpose computer, any comparable means, or the like. In general, any device(s) or means capable of implementing the methodology illustrated herein can be used to implement the various aspects of this disclosure. Exemplary hardware that can be used for the disclosed embodiments, configurations and aspects includes computers, handheld devices, telephones (e.g., cellular, Internet enabled, digital, analog, hybrids, and others), and other hardware known in the art. Some of these devices include processors (e.g., a single or multiple microprocessors), memory, nonvolatile storage, input devices, and output devices. Furthermore, alternative software implementations including, but not limited to, distributed processing or component/object distributed processing, parallel processing, or virtual machine processing can also be constructed to implement the methods described herein.

[0121] In yet another embodiment, the disclosed methods may be readily implemented in conjunction with software using object or object-oriented software development environments that provide portable source code that can be used on a variety of computer or workstation platforms. Alternatively, the disclosed system may be implemented partially or fully in hardware using standard logic circuits or VLSI design. Whether software or hardware is used to implement the systems in accordance with this disclosure is dependent on the speed and/or efficiency requirements of the system, the particular function, and the particular software or hardware systems or microprocessor or microcomputer systems being utilized.

[0122] In yet another embodiment, the disclosed methods may be partially implemented in software that can be stored on a storage medium, executed on programmed general-purpose

computer with the cooperation of a controller and memory, a special purpose computer, a microprocessor, or the like. In these instances, the systems and methods of this disclosure can be implemented as program embedded on personal computer such as an applet, JAVA® or CGI script, as a resource residing on a server or computer workstation, as a routine embedded in a dedicated measurement system, system component, or the like. The system can also be implemented by physically incorporating the system and/or method into a software and/or hardware system.

[0123] Although the present disclosure describes components and functions implemented in the aspects, embodiments, and/or configurations with reference to particular standards and protocols, the aspects, embodiments, and/or configurations are not limited to such standards and protocols. Other similar standards and protocols not mentioned herein are in existence and are considered to be included in the present disclosure. Moreover, the standards and protocols mentioned herein and other similar standards and protocols not mentioned herein are periodically superseded by faster or more effective equivalents having essentially the same functions. Such replacement standards and protocols having the same functions are considered equivalents included in the present disclosure.

[0124] The present disclosure, in various aspects, embodiments, and/or configurations, includes components, methods, processes, systems and/or apparatus substantially as depicted and described herein, including various aspects, embodiments, configurations embodiments, subcombinations, and/or subsets thereof. Those of skill in the art will understand how to make and use the disclosed aspects, embodiments, and/or configurations afrer understanding the present disclosure. The present disclosure, in various aspects, embodiments, and/or configurations, includes providing devices and processes in the absence of items not depicted and/or described herein or in various aspects, embodiments, and/or configurations hereof, including in the absence of such items as may have been used in previous devices or processes, e.g., for improving performance, achieving ease and\or reducing cost of implementation.

[0125] The foregoing discussion has been presented for purposes of illustration and description. The foregoing is not intended to limit the disclosure to the form or forms disclosed herein. In the foregoing Detailed Description for example, various features of the disclosure are grouped together in one or more aspects, embodiments, and/or configurations for the purpose of streamlining the disclosure. The features of the aspects, embodiments, and/or configurations of

the disclosure may be combined in alternate aspects, embodiments, and/or configurations other than those discussed above. This method of disclosure is not to be interpreted as reflecting an intention that the claims require more features than are expressly recited in each claim. Rather, as the following claims reflect, inventive aspects lie in less than all features of a single foregoing disclosed aspect, embodiment, and/or configuration. Thus, the following claims are hereby incorporated into this Detailed Description, with each claim standing on its own as a separate preferred embodiment of the disclosure.

[0126] Moreover, though the description has included description of one or more aspects, embodiments, and/or configurations and certain variations and modifications, other variations, combinations, and modifications are within the scope of the disclosure, e.g., as may be within the skill and knowledge of those in the art, after understanding the present disclosure. It is intended to obtain rights which include alternative aspects, embodiments, and/or configurations to the extent permitted, including alternate, interchangeable and/or equivalent structures, functions, ranges or steps to those claimed, whether or not such alternate, interchangeable and/or equivalent structures, functions, ranges or steps are disclosed herein, and without intending to publicly dedicate any patentable subject matter.

What is claimed is:

1. A method of displaying an application onto a desktop on a multi-screen device, comprising:

displaying a first application on a first screen of the multi-screen device;

receiving a first predetermined input to display a desktop on the first screen and the second screen of the multi-screen device;

responding to the first predetermined input with an output that causes the desktop to be displayed on the first screen and the second screen of the multi-screen device;

receiving a second predetermined input on the first screen that represents an instruction to display a second application on the first screen of the multi-screen device;

responding to the second predetermined input with an output that causes the second application to be displayed on the first screen and which also causes the first application to be displayed on the second screen of the multi-screen device.

2. The method of claim 1, wherein in the first application is a multi-screen application and is displayed simultaneously on the first and second screen.

3. The method of claim 2, wherein in response to the first predetermined input the multi-screen application is stored in a virtual stack, the method further comprising:

registering, in memory, the first application for display by the first screen and the second screen;

preventing the first application from displaying on the first screen and the second screen where the first desktop screen and second desktop screen is displayed; and

wherein access to the memory is provided such that the first application can be retrieved.

4. The method of claim 3, wherein the virtual stack is split into a left stack and a right stack.

5. The method of claim 4, wherein the application is split into a first and second part where the first part is stored in a first stack and the second part is stored in a second stack; and

wherein the first stack represents data displayed on the first screen and the second stack represents data displayed on the second screen; and

wherein the first stack is stored in the left stack and the second stack is stored in the right stack.

6. The method of claim 1, wherein in response to the second predetermined input the first application is displayed on the second screen in a single-page format.

7. The method of claim 1, wherein in the first and second applications are the same.

8. The method of claim 1, wherein prior to receiving a first predetermined input to display a desktop on the first screen and the second screen of the multi-screen device, the multi-screen device is displaying the first application on the first screen and a third application on the second screen; and

wherein responding to the second predetermined input with an output that causes the second application to be displayed on the first screen also causes the determination of at least one of 1) the first application to be displayed on the second screen, and 2) the third application to be displayed on the second screen.

9. The method of claim 8, wherein in the determination of displaying a particular application involves utilizing at least one of 1) application priority information; 2) the application sequence; 3) a logical display algorithm based on predefined application criteria.

10. A non-transitory computer-readable medium having stored thereon instructions that cause a computing system to execute a method, the instructions comprising:

instructions configured to display a first application on a first screen of a multi-screen device;

instructions configured to receive a first predetermined input that represents an instruction to reveal a desktop on the first screen and the second screen of the multi-screen device;

instructions configured to respond to the first predetermined input with an output that causes the desktop to be displayed on the first screen and the second screen of the multi-screen device;

instructions configured to receive a second predetermined input that represents an instruction to display a second application on the first screen of the multi-screen device;

instructions configured to respond to the second predetermined input with an output that causes the second application to be displayed on the first screen and which also causes the first application to be displayed on the second screen of the multi-screen device.

11. The computer-readable medium of claim 10, wherein the first screen corresponds to a primary screen and wherein the second screen corresponds to a secondary screen.

12. The computer-readable medium of claim 10, wherein in the first application is a multi-screen application and is displayed simultaneously on the first and second screen.

13. The method of claim 10, wherein in response to the first predetermined input the multi-screen application is stored in a virtual stack, the method further comprising:

registering, in memory, the first application for display by the first screen and the second screen;

preventing the first application from displaying on the first screen and the second screen where the first desktop screen and second desktop screen is displayed; and

wherein access to the memory is provided such that the first application can be retrieved.

14. The computer-readable medium of claim 13, wherein the virtual stack is split into a left stack and a right stack.

15. The method of claim 14, wherein the application is split into a first and second part where the first part is stored in a first stack and the second part is stored in a second stack; and

wherein the first stack represents data displayed on the first screen and the second stack represents data displayed on the second screen; and

wherein the first stack is stored in the left stack and the second stack is stored in the right stack.

16. The computer-readable medium of claim 10, wherein in response to the second predetermined input the first application is displayed on the second screen in a single-page format.

17. The computer-readable medium of claim 10, wherein the first and second applications are the same.

18. The method of claim 10, wherein prior to receiving a first predetermined input to display a desktop on the first screen and the second screen of the multi-screen device, the multi-screen device is displaying the first application on the first screen and a third application on the second screen; and

wherein responding to the second predetermined input with an output that causes the second application to be displayed on the first screen also causes the determination of at least one of 1) the first application to be displayed on the second screen, and 2) the third application to be displayed on the second screen.

19. The method of claim 18, wherein in the determination of displaying a particular application involves utilizing at least one of 1) application priority information; 2) the application sequence; and 3) a logical display algorithm based on predefined application criteria.

20. A dual-screen user device, comprising:

a first screen including a first display area;

a second screen including a second display area;

a first user input gesture area configured to receive a first signal indicative of a position on the first screen;

a second user input gesture area configured to receive a second signal indicative of a position on the second screen; and

a computer-readable medium having instructions stored thereon that include:

a first set of instructions configured to determine a received signal based on the input origin of a first or second signal; and

a second set of instructions configured to determine whether the received signal corresponds to the position of a known application displayed on the first or second display; and

a third set of instructions configured to launch the known application and display it to the screen from where the received signal originated.

ABSTRACT

A dual-screen user device and methods for launching applications from a revealed desktop onto a logically chosen screen are disclosed. Specifically, a user reveals the desktop and then launches a selected application from one of two desktops displayed on a primary and secondary screen of a device. When the application is launched, it is displayed onto a specific screen depending on the input received and the logical rules determining the display output. As the application is displayed onto the specific screen, the desktop is removed from display and the opposite screen can display other data.

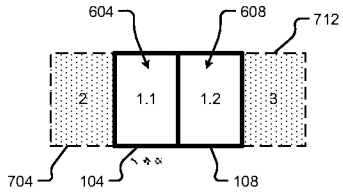


FIG. 7A

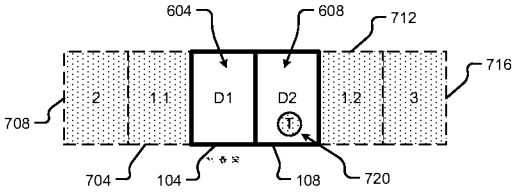


FIG. 7B

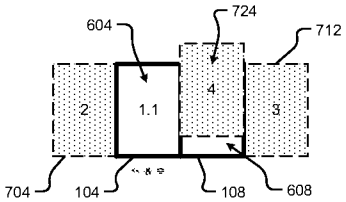


FIG. 7C

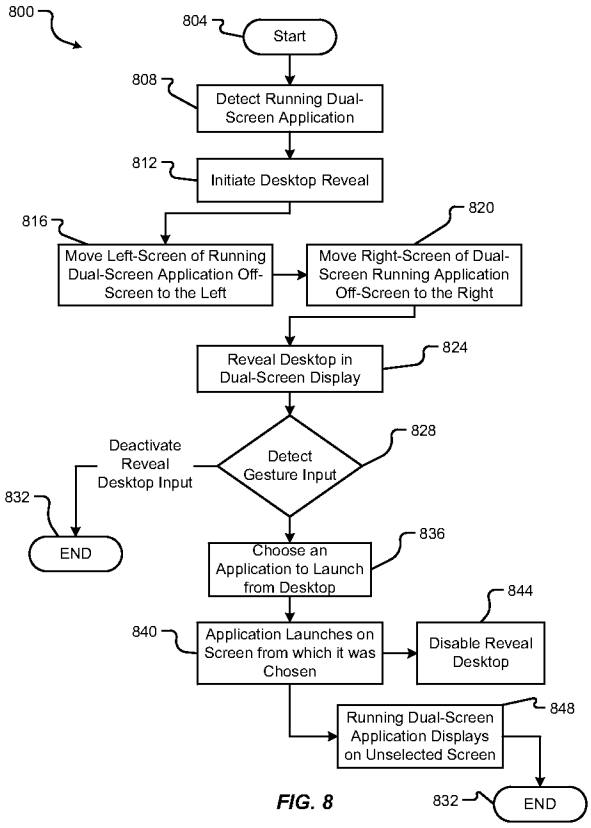


FIG. 8

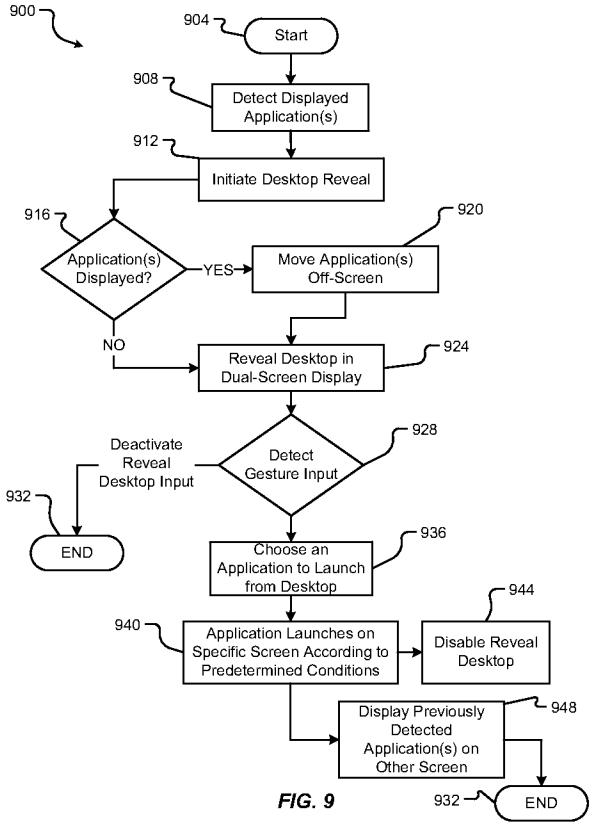


FIG. 9

DUAL SCREEN APPLICATION BEHAVIOUR

[0100] Display Controls:

[0101] FIGS. 7A-7D depict graphical representations of embodiments of a device as it passes through several different display states. Specifically, the device is shown in an open portrait state 320 and a closed portrait state 304. If a screen is shown surrounded by solid lines, then that particular screen is displaying data to the screen. Pages that are not currently visible from the primary screen 104 or secondary screen 108 are represented by surrounding dashed lines.

[0102] In one embodiment of the device 100, the MDM class 524 arranges applications, desktops, and/or other displayable information through the Window Management class 532 onto separate pages 704, 708, 712, 716 and organizes those pages 704, 708, 712, 716 into virtual "stacks." These stacks of pages can be created, deleted, shuffled, and/or moved by a user or an output from the MDM class 524, similar to a user manipulating the virtual playing cards of a deck stack. In FIG. 7A, the stacks of individual pages 704, 708, 712, 716 are shown spread outwardly in a linear format on both the left side of the primary screen 104 as well the right side of the secondary screen 108 to better visualize the organization of the virtual stacks. The position of a page from the primary screen 104 or secondary screen 108 is the position of the page in each respective stack stored on the device 100. Therefore, if a page is graphically represented immediately adjacent to the primary screen 104, or immediately adjacent to the secondary screen 108, it is the first page in the stack that may be moved onto that adjacent screen with a user input gesture. Although only several pages of a stack are shown in FIG. 7A, the number of pages available to be organized in the stack can vary and, in some embodiments, may equal the number of applications, desktops, and/or other displayable information created on the device 100. Moreover, the stack adjacent to the primary screen 104 and the stack adjacent to the secondary screen 108 may act as a unified stack (where a move or manipulation of a page from one stack moves or manipulates the pages in both stacks), or as independent stacks (each screen having its own independently navigable stack of pages).

[0103] In an embodiment of the present disclosure, it is anticipated that the concept of stacks disclosed herein may also be applied to desktops available on the device 100. In other words, a desktop virtual stack may be used for the desktops which can be displayed by the device 100 and an application virtual stack may be used for the applications which can be displayed by the device 100. Specifically, when a desktop is revealed it may be divided into multiple pages for

display on multiple screens of the device 100. These pages may be manipulated in their own desktop virtual stack, or stacks, separate from the application virtual stacks and/or other virtual stacks. This separation of stacks allows for greater user flexibility in navigating through different applications or desktops by creating an intuitive interface dependent on the data displayed to the screens 104, 108. However, it is also anticipated that these stacks could be combined to form a single virtual stack.

[0104] FIG. 7A shows a graphical representation of the device 100 in an open portrait state 320 where a first dual-screen application is running and displaying data in the Pmax configuration 624 (on the primary screen 104 and secondary screen 108 simultaneously). A second application is shown running on first stack first page 704 and a third application is running on the second stack first page 712. Other pages 712, 716 that are not visible on the primary screen 104 or the secondary screen 108 are graphically represented as surrounded by dashed lines.

[0105] FIG. 7B shows an embodiment of the present device 100 in a closed portrait state 304 where the device 100 receives a first input to restrict displayed data to the primary screen 104 only, and the first dual-screen application is minimized to be displayed by the primary screen 104 in a first portrait configuration 604. The device 100 may be physically manipulated and/or closed to provide the first input necessary to display data only to the primary screen 104, but this first input could alternatively be provided by, (1) a user input gesture, (2) a combination of user input gestures, (3) a memory output, (4) a response to a predetermined condition (e.g., application control, power levels, communications interrupt, operating system state, device screen state open/closed, timers, and single or multiple sensor outputs) or (5) any combination thereof, that the MDM class 524 registers and interprets as the necessary first input.

[0106] FIG. 7C shows an embodiment of the present device 100 in an open portrait state 320 where the device 100 receives a second input that reconfigures the display of data to output to both the primary screen 104 and secondary screen 108. Similar to the input required to restrict the display of data, the device 100 may be physically manipulated and/or closed to provide the second input necessary to display data to both screens 104, 108, but this second input could alternatively be provided by, (1) a user input gesture, (2) a combination of user input gestures, (3) a memory output, (4) a response to a predetermined condition (e.g., application control, power levels, communications interrupt, operating system state, device screen state open/closed,

timers, and single or multiple sensor outputs) or (5) any combination thereof, that the MDM class 524 registers and interprets as the necessary first input.

[0107] It is one aspect of the present disclosure that afrer returning from a closed portrait state 304, as described above (and shown in FIG. 7B), the first dual-screen application that was minimized to the primary screen 104 is not automatically maximized to run on the primary and secondary screens 104, 108. If the first dual-screen application is not maximized to display on both screens 104, 108, the secondary screen 108 displays the data stored on the second stack first page in a second portrait configuration 608. This data could include, but is not limited to, a desktop, a home page, an application, or a combination thereof. In this example, because the third application was stored on the second stack first page 712 the third application is displayed by the secondary screen 108 in a second portrait configuration 608.

[0108] FIG. 7D shows an embodiment of the present device 100 in a closed portrait state 304 where the device 100 receives the first input to restrict displayed data to the primary screen 104 only, and where the first dual-screen application is shown running, minimized, in a first portrait configuration 604 while the data previously displayed to the secondary screen 108, the third application in this example, is returned to the second stack first page 712. The secondary screen 108 is not displaying data to the screen 108.

[0109] FIG. 8A shows a graphical representation of the device 100 in an open portrait state 320 where a modal dual-screen application is running and displaying data in the Pmax configuration 624 (on the primary screen 104 and secondary screen 108 simultaneously). A second application is shown running on first stack first page 704 and a third application is running on the second stack first page 712. Other pages 712, 716 that are not visible on the primary screen 104 or the secondary screen 108 are graphically represented as surrounded by dashed lines.

[0110] FIG. 8B shows an embodiment of the present device 100 in a closed portrait state 304 where the device 100 receives a first input to restrict displayed data to the primary screen 104 only, and the modal dual-screen application is minimized to be displayed by the primary screen 104 in a first portrait configuration 604. The device 100 may be physically manipulated and/or closed to provide the first input necessary to display data only to the primary screen 104, but this first input could alternatively be provided by, (1) a user input gesture, (2) a combination of user input gestures, (3) a memory output, (4) a response to a predetermined condition (e.g.,

application control, power levels, communications interrupt, operating system state, device screen state open/closed, timers, and single or multiple sensor outputs) or (5) any combination thereof, that the MDM class 524 registers and interprets as the necessary first input.

[0111] FIG. 8C shows an embodiment of the present device 100 in an open portrait state 320 where the device 100 receives a second input that reconfigures the display of data to output to both the primary screen 104 and secondary screen 108. Similar to the input required to restrict the display of data, the device 100 may be physically manipulated and/or closed to provide the second input necessary to display data to both screens 104, 108, but this second input could alternatively be provided by, (1) a user input gesture, (2) a combination of user input gestures, (3) a memory output, (4) a response to a predetermined condition (e.g., application control, power levels, communications interrupt, operating system state, device screen state open/closed, timers, and single or multiple sensor outputs) or (5) any combination thereof, that the MDM class 524 registers and interprets as the necessary first input.

[0112] It is one aspect of the present disclosure that afrer returning from a closed portrait state 304 to an open portrait state 320, as described above (and shown in FIG. 8B), that modal minimization rules determine whether the modal dual-screen application is maintained as a minimized application displayed to the primary screen 104, or if it is maximized to run on the primary and secondary screens 104, 108. These modal minimization rules allow the MDM class 524 to determine how specific applications, data, desktops, or other information is displayed, or directed not to display, on one or more of the primary screen 104, secondary screen 108, and both screens 104, 108. More specifically, the decision of how to display modal controlled data, including applications, may depend on the application priority, the sequence of applications created, predetermined display preferences, user-set preferences, and/or other determinative functions. If the modal minimization rules direct the modal dual-screen application to display in a PMax configuration 624 (FIG. 8A), then the modal dual-screen application is maximized to display on the primary and secondary screens 104, 108. On the other hand, if the modal dual-screen application is not maximized to display on both screens 104, 108, the secondary screen 108 displays the data stored on the second stack first page in a second portrait configuration 608. This data could include, but is not limited to, a desktop, a home page, an application, or a combination thereof. In this example, because the third application was stored on the second stack first page 712 the third application is displayed by the secondary screen 108 in a second

portrait configuration 608. Although explained here in the context of a running application, this disclosure relates to any information capable of being displayed to at least one of the primary screen 104, the secondary screen 108, and both screens 104, 108.

[0113] FIG. 8D shows an embodiment of the present device 100 in a closed portrait state 304 where the device 100 receives the first input to restrict displayed data to the primary screen 104 only, and where the modal dual-screen application is shown running, minimized, in a first portrait configuration 604 while data stored on the second stack first page 712 remains stored on the second stack first page 712. The secondary screen 108 is not displaying data to the screen 108.

[0114] FIG. 9 is a flow diagram depicting an aspect of dual-screen application behavior in accordance with embodiments of the present disclosure. The method is initiated at step 904. In embodiments, a device 100 may be running a first dual-screen application displayed to the primary screen 104 and secondary screen 108 simultaneously in a maximized mode when one of the displays is deactivated (step 912). It is anticipated that a screen of the device 100 may be deactivated manually or automatically in response to specific conditions. One example of a manual deactivation of a screen may occur when the device 100 is closed from an open state, where both screens were actively displaying data, to a closed state where only one screen is actively displaying data. Upon the deactivation of a screen, for the sake of example the secondary screen 108, the previously displayed and maximized first dual-screen application is minimized to display on the primary screen 104 in a single-screen state (step 916). Once the deactivated screen is reactivated (i.e., actively displaying data to the screen) the device 100 must determine the data to display to the reactivated screen (step 920). In some embodiments, the first dual-screen application is not maximized to display on the reactivated screen. Instead, the minimized first dual-screen application may remain displayed on the primary screen 104 (step 928) while the secondary screen 108 is free to display other data. It is anticipated that the data displayed on the reactivated screen may be a second application (step 932). The method ends at step 936.

[0115] Referring now to FIG. 10 a flow diagram depicting an aspect of dual-screen application behavior is shown in accordance with embodiments of the present disclosure. The method is initiated at step 1004. In embodiments, a device 100 may be running a first dual-screen application displayed to the primary screen 104 and secondary screen 108 simultaneously in a

maximized mode when one of the displays is deactivated (step 1012). It is anticipated that a screen of the device 100 may be deactivated manually or automatically as described above. Upon detecting the deactivation of a screen, for the sake of example the secondary screen 108, the previously displayed and maximized first dual-screen application is minimized to display on the primary screen 104 in a single-screen state (step 1016). Once the deactivated screen is reactivated (i.e., actively displaying data to the screen) the device 100 must determine the data to display to the reactivated screen (step 1020-1024).

[0116] FIG. 10 shows an embodiment of the present disclosure where determining the data to display to the screens may depend on specific maximization and/or minimization rules, also known as modal rules (step 1024). If no modal rules are defined for a given application, the default condition is such that the first dual-screen application is not maximized to display on the reactivated screen. In fact, the minimized first dual-screen application may remain displayed on the primary screen 104 (step 1028) while the secondary screen 108 is free to display other data (including a second application). In the event that modal rules are defined for a given application, the first dual-screen application may be maximized and/or minimized according to the modal rules. The method ends at step 936.

[0117] The exemplary systems and methods of this disclosure have been described in relation to display controls for a multi-screen device. However, to avoid unnecessarily obscuring the present disclosure, the preceding description omits a number of known structures and devices. This omission is not to be construed as a limitation of the scopes of the claims. Specific details are set forth to provide an understanding of the present disclosure. It should however be appreciated that the present disclosure may be practiced in a variety of ways beyond the specific detail set forth herein.

[0118] Furthermore, while the exemplary aspects, embodiments, and/or configurations illustrated herein show the various components of the system collocated, certain components of the system can be located remotely, at distant portions of a distributed network, such as a LAN and/or the Internet, or within a dedicated system. Thus, it should be appreciated, that the components of the system can be combined in to one or more devices, such as a Personal Computer (PC), laptop, netbook, Personal Digital Assistant (PDA), tablet, etc., or collocated on a particular node of a distributed network, such as an analog and/or digital telecommunications network, a packet-switch network, or a circuit-switched network. It will be appreciated from the

preceding description, and for reasons of computational efficiency, that the components of the system can be arranged at any location within a distributed network of components without affecting the operation of the system. For example, the various components can be located in a switch such as a PBX and media server, gateway, in one or more communications devices, at one or more users' premises, or some combination thereof. Similarly, one or more functional portions of the system could be distributed between a telecommunications device(s) and an associated computing device.

[0119] Furthermore, it should be appreciated that the various links connecting the elements can be wired or wireless links, or any combination thereof, or any other known or later developed element(s) that is capable of supplying and/or communicating data to and from the connected elements. These wired or wireless links can also be secure links and may be capable of communicating encrypted information. Transmission media used as links, for example, can be any suitable carrier for electrical signals, including coaxial cables, copper wire and fiber optics, and may take the form of acoustic or light waves, such as those generated during radio-wave and infra-red data communications.

[0120] Also, while the flowcharts have been discussed and illustrated in relation to a particular sequence of events, it should be appreciated that changes, additions, and omissions to this sequence can occur without materially affecting the operation of the disclosed embodiments, configuration, and aspects.

[0121] A number of variations and modifications of the disclosure can be used. It would be possible to provide for some features of the disclosure without providing others.

[0122] In some embodiments, the systems and methods of this disclosure can be implemented in conjunction with a special purpose computer, a programmed microprocessor or microcontroller and peripheral integrated circuit element(s), an ASIC or other integrated circuit, a digital signal processor, a hard-wired electronic or logic circuit such as discrete element circuit, a programmable logic device or gate array such as PLD, PLA, FPGA, PAL, special purpose computer, any comparable means, or the like. In general, any device(s) or means capable of implementing the methodology illustrated herein can be used to implement the various aspects of this disclosure. Exemplary hardware that can be used for the disclosed embodiments, configurations and aspects includes computers, handheld devices, telephones (e.g., cellular, Internet enabled, digital, analog, hybrids, and others), and other hardware known in the art.

Some of these devices include processors (e.g., a single or multiple microprocessors), memory, nonvolatile storage, input devices, and output devices. Furthermore, alternative software implementations including, but not limited to, distributed processing or component/object distributed processing, parallel processing, or virtual machine processing can also be constructed to implement the methods described herein.

[0123] In yet another embodiment, the disclosed methods may be readily implemented in conjunction with software using object or object-oriented software development environments that provide portable source code that can be used on a variety of computer or workstation platforms. Alternatively, the disclosed system may be implemented partially or fully in hardware using standard logic circuits or VLSI design. Whether software or hardware is used to implement the systems in accordance with this disclosure is dependent on the speed and/or efficiency requirements of the system, the particular function, and the particular software or hardware systems or microprocessor or microcomputer systems being utilized.

[0124] In yet another embodiment, the disclosed methods may be partially implemented in software that can be stored on a storage medium, executed on programmed general-purpose computer with the cooperation of a controller and memory, a special purpose computer, a microprocessor, or the like. In these instances, the systems and methods of this disclosure can be implemented as program embedded on personal computer such as an applet, JAVA® or CGI script, as a resource residing on a server or computer workstation, as a routine embedded in a dedicated measurement system, system component, or the like. The system can also be implemented by physically incorporating the system and/or method into a software and/or hardware system.

[0125] Although the present disclosure describes components and functions implemented in the aspects, embodiments, and/or configurations with reference to particular standards and protocols, the aspects, embodiments, and/or configurations are not limited to such standards and protocols. Other similar standards and protocols not mentioned herein are in existence and are considered to be included in the present disclosure. Moreover, the standards and protocols mentioned herein and other similar standards and protocols not mentioned herein are periodically superseded by faster or more effective equivalents having essentially the same functions. Such replacement standards and protocols having the same functions are considered equivalents included in the present disclosure.

[0126] The present disclosure, in various aspects, embodiments, and/or configurations, includes components, methods, processes, systems and/or apparatus substantially as depicted and described herein, including various aspects, embodiments, configurations embodiments, subcombinations, and/or subsets thereof. Those of skill in the art will understand how to make and use the disclosed aspects, embodiments, and/or configurations after understanding the present disclosure. The present disclosure, in various aspects, embodiments, and/or configurations, includes providing devices and processes in the absence of items not depicted and/or described herein or in various aspects, embodiments, and/or configurations hereof, including in the absence of such items as may have been used in previous devices or processes, e.g., for improving performance, achieving ease and\or reducing cost of implementation.

[0127] The foregoing discussion has been presented for purposes of illustration and description. The foregoing is not intended to limit the disclosure to the form or forms disclosed herein. In the foregoing Detailed Description for example, various features of the disclosure are grouped together in one or more aspects, embodiments, and/or configurations for the purpose of streamlining the disclosure. The features of the aspects, embodiments, and/or configurations of the disclosure may be combined in alternate aspects, embodiments, and/or configurations other than those discussed above. This method of disclosure is not to be interpreted as reflecting an intention that the claims require more features than are expressly recited in each claim. Rather, as the following claims reflect, inventive aspects lie in less than all features of a single foregoing disclosed aspect, embodiment, and/or configuration. Thus, the following claims are hereby incorporated into this Detailed Description, with each claim standing on its own as a separate preferred embodiment of the disclosure.

[0128] Moreover, though the description has included description of one or more aspects, embodiments, and/or configurations and certain variations and modifications, other variations, combinations, and modifications are within the scope of the disclosure, e.g., as may be within the skill and knowledge of those in the art, after understanding the present disclosure. It is intended to obtain rights which include alternative aspects, embodiments, and/or configurations to the extent permitted, including alternate, interchangeable and/or equivalent structures, functions, ranges or steps to those claimed, whether or not such alternate, interchangeable and/or equivalent structures, functions, ranges or steps are disclosed herein, and without intending to publicly dedicate any patentable subject matter.

What is claimed is:

1. A method of controlling the size of a dual-screen application displayed on a multi-screen device, comprising:

displaying a dual-screen application on a first screen and a second screen of the multi-screen device;

receiving a first predetermined input that represents an instruction to deactivate displaying to the second screen of the multi-screen device;

responding to the first predetermined input with an output that deactivates the second screen display and causes the dual-screen application to be displayed in a single-screen mode on the first screen of the multi-screen device;

receiving a second predetermined input that represents an instruction to reactivate displaying to the second screen of the multi-screen device;

responding to the second predetermined input with an output that reactivates the second screen display and causes the dual-screen application to continue to be displayed in a single-screen mode on the first screen of the multi-screen device.

2. The method of claim 1, wherein in the first predetermined input is caused by a physical change of state to the multi-screen device.

3. The method of claim 2, wherein in the second predetermined input is caused by a physical change of state to the multi-screen device.

4. The method of claim 3, wherein the physical change of state is effected by at least one of opening the multi-screen device and closing the multi-screen device.

5. The method of claim 1, wherein the multi-screen device is running a second application stored in a virtual stack.

6. The method of claim 5, wherein after receiving the second predetermined input, the second application is displayed to the second screen of the multi-screen device.

7. The method of claim 1, wherein responding to the second predetermined input further includes referring to modal rules of the dual-screen application, wherein the modal rules define the manner in which the dual-screen application is displayed to the first screen.

8. The method of claim 7, wherein the modal rules define that the dual-screen application is maximized to display data to the first and second screens.

9. The method of claim 7, wherein the modal rules define that the dual-screen application is displayed to the first screen.

10. A non-transitory computer-readable medium having stored thereon instructions that cause a computing system to execute a method, the instructions comprising:

instructions configured to display a dual-screen application on a first screen and a second screen of the multi-screen device;

instructions configured to receive a first predetermined input that represents an instruction to deactivate displaying to the second screen of the multi-screen device;

instructions configured to respond to the first predetermined input with an output that deactivates the second screen display and causes the dual-screen application to be displayed in a single-screen mode on the first screen of the multi-screen device;

instructions configured to receive a second predetermined input that represents an instruction to reactivate displaying to the second screen of the multi-screen device;

instructions configured to respond to the second predetermined input with an output that reactivates the second screen display and causes the dual-screen application to continue to be displayed in a single-screen mode on the first screen of the multi-screen device.

11. The computer-readable medium of claim 10, wherein in the first predetermined input is caused by a physical change of state to the multi-screen device.

12. The computer-readable medium of claim 11, wherein in the second predetermined input is caused by a physical change of state to the multi-screen device.

13. The computer-readable medium of claim 12, wherein the physical change of state is effected by at least one of opening the multi-screen device and closing the multi-screen device.

14. The computer-readable medium of claim 10, wherein the multi-screen device is running a second application stored in a virtual stack.

15. The computer-readable medium of claim 14, wherein after receiving the second predetermined input, the second application is displayed to the second screen of the multi-screen device.

16. The computer-readable medium of claim 10, wherein responding to the second predetermined input further includes referring to modal rules of the dual-screen application, wherein the modal rules define the manner in which the dual-screen application is displayed to the first screen.

17. The computer-readable medium of claim 16, wherein the modal rules define that the dual-screen application is maximized to display data to the first and second screens.

18. The computer-readable medium of claim 16, wherein the modal rules define that the dual-screen application is displayed to the first screen.

19. A dual-screen user device, comprising:

a first screen including a first display area;

a second screen including a second display area;

a hinge configured to connect the first screen with the second screen;

a first sensor configured to produce a first signal indicative of a position of the hinge;

the computer-readable medium of claim 10, including further instructions configured to determine, based at least partially on the first signal, that the hinge has moved ftom a first position in which the first and second screens are open to a second position in which the first and second screens are closed.

20. The device of claim 19, wherein the second position is the first predetermined input, and first position is the second predetermined input.

ABSTRACT

A multi-screen user device and methods for logically controlling the display behavior of applications and other displayable data are disclosed. Specifically, when a dual-screen application is displayed to the primary and secondary screens of a device the dual-screen application can be minimized to a single screen by closing the device. When the device is reopened, the minimized dual-screen application is maintained in a minimized state on the screen to which it was minimized. Specific rules can be implemented to determine whether the dual-screen application will be maximized to run on the primary and secondary screens of a device upon reopening.

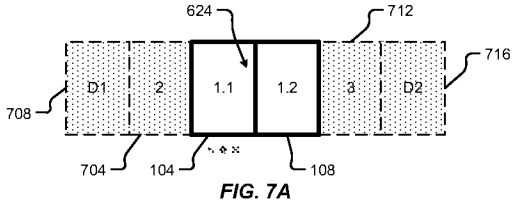


FIG. 7A

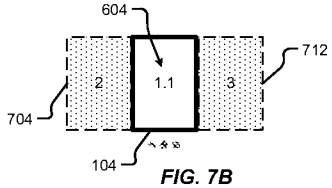


FIG. 7B

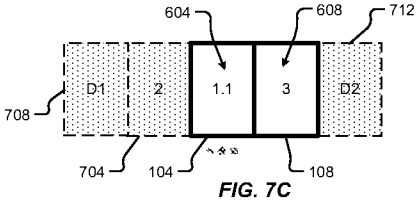


FIG. 7C

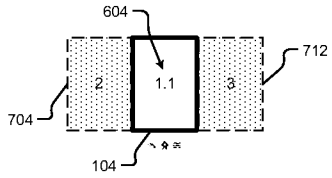


FIG. 7D

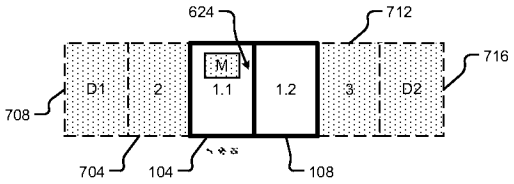


FIG. 8A

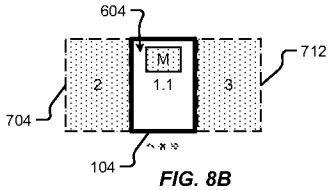


FIG. 8B

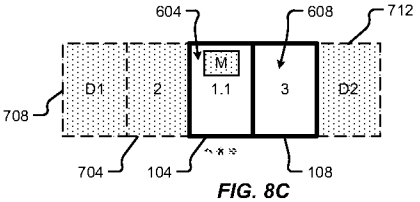


FIG. 8C

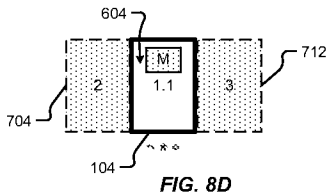


FIG. 8D

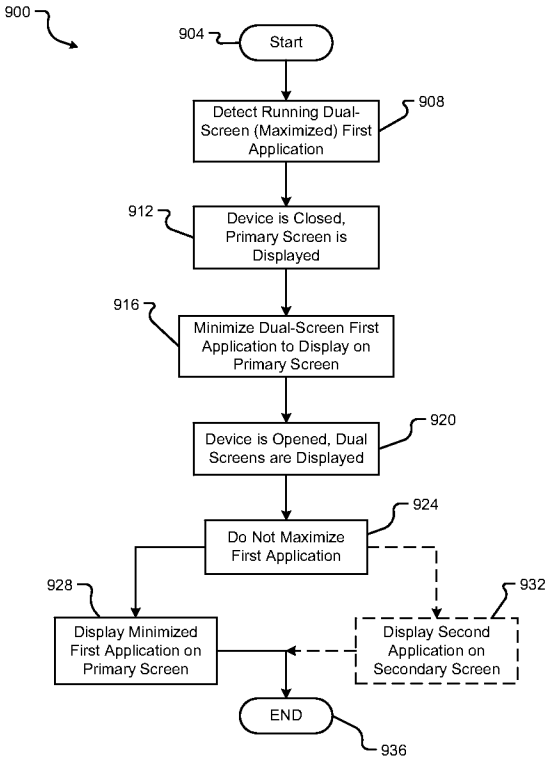


FIG. 9

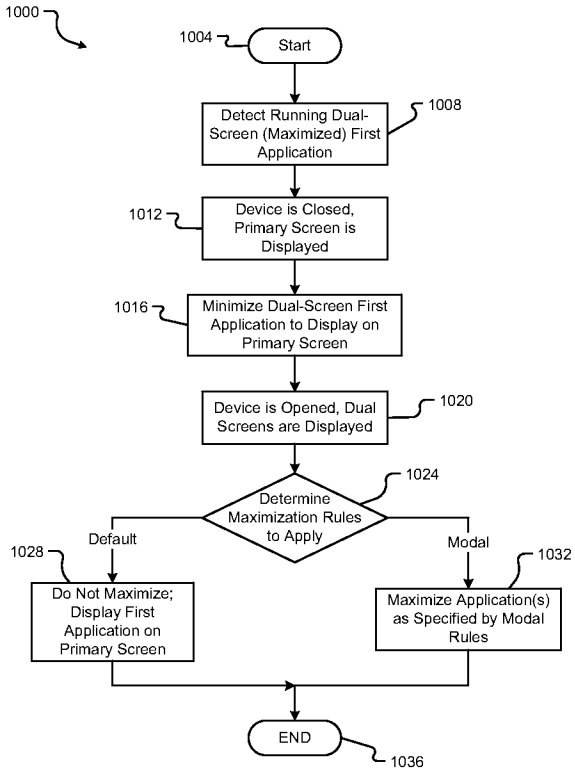


FIG. 10

GRAVITY DROP

[0001] Display Controls:

[0002] Figs. 7A-C depict three examples of gesture sequences that can cause the device 100 to alter the display of data thereon. Specifically, Fig. 7A depicts a first gesture sequence 700 which results in a first display action. Although the first display action is depicted as being performed on the touch sensitive displays 110, 114, it should be appreciated that the first display action can alternatively, or additionally, be performed on other parts of the primary screen 104 and/or secondary screen 108. The first gesture sequence 700 begins when the device 100 is in a closed landscape state 340. While in this state, the device 100 is displaying data in a first landscape configuration 612. The user of the device 100 then provides a first user input 704 causing the device 100 to change from the closed landscape state 340 to an open landscape state 348. This first user input 704 may be detected by one or more of the hardware components of the device 100 as discussed herein.

[0003] In the example of Fig. 7A, the data displayed in the first landscape configuration 612 may be derived from a single-screen application, which means that the application data currently being displayed on the touch sensitive display 110 is originating from an application that is configured to only display data on a single screen 104 or 108. Accordingly, either before or afrer the MDM module 524 registers the first user input 704, the MDM module 524 determines that the application being displayed in the first landscape configuration 612 corresponds to a single-screen application. Thus, in response to the first user input 704, the MDM module 524 instructs the window management module 532 to change the display configuration of the device 100. Specifically, the MDM module 524 may provide instructions to display data from a second application (different from the first application being displayed on the touch sensitive display 110) on the touch sensitive display 114. Alternatively, or in addition, the window management module 532 may make the determination as to which applications should be displayed. Specifically, since the application being displayed on the touch sensitive display 110 corresponds to a single-screen application, the MDM module 524 may be configured to select a second application and cause data therefrom to be displayed in a second landscape configuration 616 via touch sensitive display 114.

[0004] In some embodiments, the second application is selected by the window management module 532 may correspond to a next application behind the first application in the window stack 550. More specifically, as part of selecting the second application, the MDM module 524 and/or window management module 532 may reference the window stack 550 that identifies active and non-active applications, as well as display priorities for such applications and then select the second application as the next application in the window stack 550. If no application is "next" behind the first application, then the window management module 532 may cause either the desktop or some other default display element to be displayed on the touch sensitive display 114.

[0005] Fig. 7B depicts a second gesture sequence 708, which results in a second display action. Similar to the first display action, the second display action can be performed on any visualization or output component of the device 100. The second gesture sequence 708 begins similarly to the first gesture sequence 700 with the device 100 being in a closed landscape state 340. The difference between the two sequences, however, is that the application being displayed on the touch sensitive display 110 is derived from a multi-screen application. While the device 100 is in the closed landscape state 340, first data from the first application is displayed via the touch sensitive display 110. As indicated by the dotted lines under touch sensitive display 110, the other touch sensitive display 114 is not actively displaying any data.

[0006] The second gesture sequence 708 continues when a first user input 712 causing the device 100 to change from the closed landscape state 340 to an open landscape state 348. This first user input 712 may be similar or identical to the first user input 704.

[0007] Before or after the MDM module 524 registers the first user input 512, the MDM module 524 determines that the application being displayed in the first landscape configuration 612 corresponds to a multi-screen application. Thus, in response to the first user input 512, the MDM module 524 instructs the window management module 532 to begin dropping display data from the first application across the touch sensitive display 114 in a transition area 716. In some embodiments, the transition area 716 moves across the touch sensitive display 114 (e.g., in a downward motion that progressively fills more and more of the touch sensitive display 114 with data from the first application). The motion of the transition area 716 may be configured to emulate a dropping of the first landscape configuration 612 across the touch sensitive display 114 under the force of gravity.

[0008] Afrer the transition area 716 has completely moved across the touch sensitive display 114, the MDM module 524 may register a second event 720 that signifies completion of the

gravity-based drop of the first application from the touch sensitive display 110. At this point, data from the first application may be displayed by both touch sensitive displays 110, 114 in any number of display configurations. In some embodiments, the application data from the first application is automatically maximized. This may cause the first application to either be simultaneously displayed on the touch sensitive screens 110, 114 in separate landscape configurations 612, 616 (e.g., an LD output configuration) or in the Lmax configuration 628. The application itself of the MDM module 524 may control the manner in which the first application is displayed and whether the application data is displayed in an LD output configuration or an Lmax configuration 628.

[0009] Moreover, the type of data displayed from the first application on the touch sensitive displays 110, 114 may vary depending upon the nature of the first application. As a non-limiting example, if the type of user input available for interfacing with the first application may change from a first type of user input (e.g., a writing-capture area, a half QWERTY keyboard) to a second type of user input (e.g., a full QWERTY keyboard or combination writing-capture area and half QWERTY keyboard) when the first application is automatically maximized across the touch sensitive displays 110, 114 during the gravity drop. Likewise, the type of user output may vary from a first type of user output to a second type of user output when the first application is automatically maximized across the touch sensitive displays 110, 114 during the gravity drop.

[0010] Fig. 7C depicts a third gesture sequence 724 which results in a third display action. Similar to the first and second display actions, the third display action can be performed on any visualization or output component of the device 100. The third gesture sequence 724 begins similarly to the other gesture sequences 700, 708 with the device 100 being in a closed landscape state 340. While the device 100 is in the closed landscape state 340, a first set of data from the first application is displayed via the touch sensitive display 110. As indicated by the dotted lines above the touch sensitive display 110, the other touch sensitive display 114 is not actively displaying any data.

[0011] The third gesture sequence 724 continues when a first user input 728 causing the device 100 to change from the closed landscape state 340 to an open landscape state 348. This first user input 728 may be similar or identical to the other user inputs 704, 712, except that the secondary screen 108 is opened above the primary screen 104. Also, afrer the first user input 728 is received, the data from the first application is already being displayed on the bottom touch

sensitive display 110 (e.g., the display 110 which is positioned below the other touch sensitive display 114). The relative positions of the displays 110, 114 or more specifically the screens 104, 108 may be determined from data provided by the accelerometer 176 or some other appropriate hardware component of the device 100.

[0012] In some embodiments, when the touch sensitive display 114 is opened above the active touch sensitive display 110, the applications in view on the touch sensitive display 110 maintain their minimized and original display configuration. Specifically, the first application being displayed on the touch sensitive display 110 may be continue to be displayed in a first landscape configuration 612 and then data from a second application may be displayed in a second landscape configuration 616 via the other touch sensitive display 114 since there is no virtual force of gravity to pull the first application down onto another screen.

[0013] The display actions performed on the touch sensitive displays 110, 114 may be controlled by the collective actions of the MDM module 524 and window management module 532 based on one or more event interrupts received from various hardware devices. Such interrupts may be registered and interpreted by the task management module 540. In selecting the data that is to be displayed via the touch sensitive display 114, the MDM module 524 and/or window management module 532 may identify the next application in a display stack as discussed above. If no application is "next" behind the first application, then the MDM module 524 and/or window management module 532 may cause either the desktop (e.g., a display rendered for the **OS**) or some other default display element to be displayed on the touch sensitive display 114.

[0014] Although the three examples of display actions were given in instances where the application was initially displayed on touch sensitive display 110, it should be appreciate that if the touch sensitive display 114 is initially displaying application data then the subsequent processing steps and display actions may be reversed. Specifically, if data from a first application is initially displayed via touch sensitive display 114 when device 100 is in a closed landscape state 340, then after the first user input 704, 712, 728 is received, the window management module 532 may manipulate the subsequent display of data on the touch sensitive display 110 depending upon the position of the touch sensitive display 110 relative to the touch sensitive display 114 and/or whether the application having data displayed is a single-screen application or a multi-screen application.

[0015] Referring now to Fig. 8, a display control method 800 will be described in accordance with at least some embodiments of the present disclosure. The method is initiated at step 804. Specifically, a device 100 may be initially in a closed landscape state 340. The method continues when the opening of the device 100 is detected (step 808). In some embodiments, the opening of the device 100 is registered as an interrupt by the task management module 540 and provided as a data input from the task management module 540 to the MDM module 524. The interrupt received at the task management module 540 may correspond to registering the relative motion of the screens 104, 108 from one or more of the position sensors 172A, 172B. The MDM module 524, even upon receiving the initial interrupt information, may wait until it determines that the device 100 has actually transitioned from the closed landscape state 340 to the open landscape state 348 (e.g., has moved through any transition state 308).

[0016] After MDM module 524 has detected that the device 100 has moved into an open landscape state 348, the method continues with the MDM module 524 instructing the window management module 532 that a device opening has occurred and a display alteration may be required. The window management module 532 may then continue by determining the application type that was being displayed before the device 100 was opened (step 812). It should be appreciated that step 812 may be performed prior to step 808 or perhaps even before the method 800 is initiated (e.g., immediately upon displaying any data in the first landscape configuration 612 or second landscape configuration 616).

[0017] After or while the MDM module 524 determines that the device 100 has been opened, the MDM module 524 may also determine the relative position of the primary screen 104 as compared to the secondary screen 108 (step 816). More specifically, the MDM module 524 determines, based on input received from one or more hardware components, whether the screen which was initially displaying application data in the closed landscape state 340 is above or below the screen which is being opened.

[0018] Depending upon the relative position of the screens 104, 108 and/or whether the application being displayed is a single-screen application or a multi-screen application, the MDM module 524 and/or window management module 532 determines what type of display action is to be performed (step 820). If the application originally displayed in the closed landscape state 340 corresponds to a single-screen application and the originally-active display is opened above the originally-inactive display, then the window management module 532 may

cause the first display action to be performed (see e.g., Fig. 7A) (step 824). Specifically, the window management module 532 may cause different applications (or an application and the desktop) to be displayed on each of the displays 110, 114.

[0019] If the application originally displayed in the closed landscape state 340 corresponds to a multi-screen application and the originally-active display is opened above the originally-inactive display, then the window management module 532 may cause the second display action to be performed (see e.g., Fig. 7B) (step 828). Specifically, the window management module may cause a single application (e.g., the originally-displayed application) to be automatically displayed on both of the displays 110, 114. The automatic display action may include an automatic maximization of the first application into either an LD output configuration or in the Lmax configuration 628.

[0020] If the originally-inactive display is opened upward (e.g., above the originally-active display), then the window management module 532 may cause the third display action to be performed (see e.g., Fig. 7C) (step 832). This display action may be performed irrespective of the nature of the application being displayed (e.g., whether the application is a single-screen application or a multi-screen application).

[0021] The exemplary systems and methods of this disclosure have been described in relation to display control mechanisms. However, to avoid unnecessarily obscuring the present disclosure, the preceding description omits a number of known structures and devices. This omission is not to be construed as a limitation of the scopes of the claims. Specific details are set forth to provide an understanding of the present disclosure. It should however be appreciated that the present disclosure may be practiced in a variety of ways beyond the specific detail set forth herein.

[0022] Furthermore, while the exemplary aspects, embodiments, and/or configurations illustrated herein show the various components of the system collocated, certain components of the system can be located remotely, at distant portions of a distributed network, such as a LAN and/or the Internet, or within a dedicated system. Thus, it should be appreciated, that the components of the system can be combined in to one or more devices, such as a computer, laptop, netbook, smart phone, mobile device, etc., or collocated on a particular node of a distributed network, such as an analog and/or digital telecommunications network, a packet-switch network, or a circuit-switched network. It will be appreciated from the preceding

description, and for reasons of computational efficiency, that the components of the system can be arranged at any location within a distributed network of components without affecting the operation of the system. For example, the various components can be located in a switch such as a PBX and media server, gateway, in one or more communications devices, at one or more users' premises, or some combination thereof. Similarly, one or more functional portions of the system could be distributed between a telecommunications device(s) and an associated computing device.

[0023] Furthermore, it should be appreciated that the various links connecting the elements can be wired or wireless links, or any combination thereof, or any other known or later developed element(s) that is capable of supplying and/or communicating data to and from the connected elements. These wired or wireless links can also be secure links and may be capable of communicating encrypted information. Transmission media used as links, for example, can be any suitable carrier for electrical signals, including coaxial cables, copper wire and fiber optics, and may take the form of acoustic or light waves, such as those generated during radio-wave and infra-red data communications.

[0024] Also, while the flowcharts have been discussed and illustrated in relation to a particular sequence of events, it should be appreciated that changes, additions, and omissions to this sequence can occur without materially affecting the operation of the disclosed embodiments, configuration, and aspects.

[0025] A number of variations and modifications of the disclosure can be used. It would be possible to provide for some features of the disclosure without providing others.

[0026] In yet another embodiment, the systems and methods of this disclosure can be implemented in conjunction with a special purpose computer, a programmed microprocessor or microcontroller and peripheral integrated circuit element(s), an ASIC or other integrated circuit, a digital signal processor, a hard-wired electronic or logic circuit such as discrete element circuit, a programmable logic device or gate array such as PLD, PLA, FPGA, PAL, special purpose computer, any comparable means, or the like. In general, any device(s) or means capable of implementing the methodology illustrated herein can be used to implement the various aspects of this disclosure. Exemplary hardware that can be used for the disclosed embodiments, configurations and aspects includes computers, handheld devices, telephones (e.g., cellular, Internet enabled, digital, analog, hybrids, and others), and other hardware known in the art.

Some of these devices include processors (e.g., a single or multiple microprocessors), memory, nonvolatile storage, input devices, and output devices. Furthermore, alternative software implementations including, but not limited to, distributed processing or component/object distributed processing, parallel processing, or virtual machine processing can also be constructed to implement the methods described herein.

[0027] In yet another embodiment, the disclosed methods may be readily implemented in conjunction with software using object or object-oriented software development environments that provide portable source code that can be used on a variety of computer or workstation platforms. Alternatively, the disclosed system may be implemented partially or fully in hardware using standard logic circuits or VLSI design. Whether software or hardware is used to implement the systems in accordance with this disclosure is dependent on the speed and/or efficiency requirements of the system, the particular function, and the particular software or hardware systems or microprocessor or microcomputer systems being utilized.

[0028] In yet another embodiment, the disclosed methods may be partially implemented in software that can be stored on a storage medium, executed on programmed general-purpose computer with the cooperation of a controller and memory, a special purpose computer, a microprocessor, or the like. In these instances, the systems and methods of this disclosure can be implemented as program embedded on personal computer such as an applet, JAVA® or CGI script, as a resource residing on a server or computer workstation, as a routine embedded in a dedicated measurement system, system component, or the like. The system can also be implemented by physically incorporating the system and/or method into a software and/or hardware system.

[0029] Although the present disclosure describes components and functions implemented in the aspects, embodiments, and/or configurations with reference to particular standards and protocols, the aspects, embodiments, and/or configurations are not limited to such standards and protocols. Other similar standards and protocols not mentioned herein are in existence and are considered to be included in the present disclosure. Moreover, the standards and protocols mentioned herein and other similar standards and protocols not mentioned herein are periodically superseded by faster or more effective equivalents having essentially the same functions. Such replacement standards and protocols having the same functions are considered equivalents included in the present disclosure.

[0030] The present disclosure, in various aspects, embodiments, and/or configurations, includes components, methods, processes, systems and/or apparatus substantially as depicted and described herein, including various aspects, embodiments, configurations embodiments, subcombinations, and/or subsets thereof. Those of skill in the art will understand how to make and use the disclosed aspects, embodiments, and/or configurations afrer understanding the present disclosure. The present disclosure, in various aspects, embodiments, and/or configurations, includes providing devices and processes in the absence of items not depicted and/or described herein or in various aspects, embodiments, and/or configurations hereof, including in the absence of such items as may have been used in previous devices or processes, e.g., for improving performance, achieving ease and\or reducing cost of implementation.

[0031] The foregoing discussion has been presented for purposes of illustration and description. The foregoing is not intended to limit the disclosure to the form or forms disclosed herein. In the foregoing Detailed Description for example, various features of the disclosure are grouped together in one or more aspects, embodiments, and/or configurations for the purpose of streamlining the disclosure. The features of the aspects, embodiments, and/or configurations of the disclosure may be combined in alternate aspects, embodiments, and/or configurations other than those discussed above. This method of disclosure is not to be interpreted as reflecting an intention that the claims require more features than are expressly recited in each claim. Rather, as the following claims reflect, inventive aspects lie in less than all features of a single foregoing disclosed aspect, embodiment, and/or configuration. Thus, the following claims are hereby incorporated into this Detailed Description, with each claim standing on its own as a separate preferred embodiment of the disclosure.

[0032] Moreover, though the description has included description of one or more aspects, embodiments, and/or configurations and certain variations and modifications, other variations, combinations, and modifications are within the scope of the disclosure, e.g., as may be within the skill and knowledge of those in the art, after understanding the present disclosure. It is intended to obtain rights which include alternative aspects, embodiments, and/or configurations to the extent permitted, including alternate, interchangeable and/or equivalent structures, functions, ranges or steps to those claimed, whether or not such alternate, interchangeable and/or equivalent structures, functions, ranges or steps are disclosed herein, and without intending to publicly dedicate any patentable subject matter.

What is claimed is:

1. A method of controlling data displayed by a dual-screen device, comprising:

determining that the dual-screen device has moved from a closed landscape state to an open landscape state;

determining a relative position of a first screen that originally displayed data from a first application when the dual-screen device was in the closed landscape state and a second screen that was not displaying data from the first application when the dual-screen device was in the closed landscape state; and

controlling data displayed on the second screen in the open landscape state based on the determined relative position of the first screen and second screen.

2. The method of claim 1, wherein the first screen is determined to be above the second screen.

3. The method of claim 2, wherein the relative position of the first and second screen is determined, at least in part, based on data received from an accelerometer.

4. The method of claim 2, further comprising:

determining whether the first application is a single-screen application or a multi-screen application; and

wherein controlling data displayed on the second screen in the open landscape state is also based on whether the first application is a single-screen application or a multi-screen application.

5. The method of claim 4, wherein the first application is a multi-screen application, the method further comprising:

automatically causing data from the first application to be displayed on the second screen in the open landscape state.

6. The method of claim 5, wherein data from the first application is displayed on both the first and second screens in the open landscape state in at least one of a dual-landscape output configuration and a landscape max output configuration.

7. The method of claim 4, wherein the first application is a single-screen application and wherein data from a second application is displayed on the second screen.

8. The method of claim 1, wherein the second screen is determined to be above the first screen.

9. The method of claim 8, wherein data from a second application is displayed on the second screen simultaneous with data from the first application being displayed on the first screen.

10. A non-transitory computer-readable medium having stored thereon instructions that cause a computing system to execute a method, the instructions comprising:

instructions configured to determine that a dual-screen device has transitioned from a closed landscape state to an open landscape state;

instructions configured to determine a relative position of a first screen of the dual-screen device and a second screen of the dual-screen device, wherein the first screen comprises a first display that originally displayed data for a first application when the dual-screen device was in the closed landscape state, and wherein the second screen comprises a second display that was not displaying data from the first application when the dual-screen device was in the closed landscape state; and

instructions configured to control data displayed on the first and second displays in the open landscape state based on the determined relative position of the first screen and second screen.

11. The computer-readable medium of claim 10, wherein the first screen corresponds to a primary screen and wherein the second screen corresponds to a secondary screen.

12. The method of claim 10, wherein the first screen is determined to be above the second screen and wherein data displayed on the second display in the open landscape state is also based on whether the first application is a single-screen application or a multi-screen application.

13. The computer-readable medium of claim 12, wherein the first application is a multi-screen application and wherein the instructions configured to control data displayed on the first and second displays is further configured to generate a transition area on the second display that moves across and progressively fills the second display with data from the first application.

14. The computer-readable medium of claim 13, wherein the data from the first application is automatically displayed on both the first and second displays in at least one of a dual landscape output configuration and a landscape max output configuration.

15. The computer-readable medium of claim 12, wherein the first application is a single-screen application and wherein data from a second application is automatically displayed on the second screen immediately afrer the device transitions to the open landscape state.

16. A dual-screen user device, comprising:

a first screen including a first display area;

a second screen including a second display area;

a hinge configured to connect the first screen with the second screen;

a first sensor configured to produce a first signal indicative of a position of the hinge;

a second sensor configured to produce a second signal indicative of a relative position of the first screen and the second screen; and

a computer-readable medium having instructions stored thereon that include:

a first set of instructions configured to determine, based at least partially on the first signal, that the hinge has moved from a first position to a second position in which the first and second screens are open;

a second set of instructions configured to determine, based on the second signal, the relative position of the first and second screens, wherein the first screen comprises a first display that originally displayed data for a first application when the dual-screen device was in the closed landscape state, and wherein the second screen comprises a second display that was not displaying data from the first application when the dual-screen device was in the closed landscape state; and

a third set of instructions configured to automatically control data displayed on the first and second displays after the hinge has opened.

17. The device of claim 16, wherein the first application is a multi-screen application and wherein the third set of instructions configured to automatically control data displayed on the first and second displays is further configured to generate a transition area on the second display that moves across and progressively fills the second display with data from the first application.

18. The device of claim 17, wherein the data from the first application is automatically displayed on both the first and second displays in a dual landscape output configuration.

19. The device of claim 17, wherein the data from the first application is automatically displayed on both the first and second displays in a landscape max output configuration.

20. The device of claim 17, wherein the first screen is determined to be above the second screen, based at least partially on the second signal.

ABSTRACT

A dual-screen user device and methods for controlling data displayed thereby are disclosed. Specifically, the data displayed by the multiple screens of the dual-screen user device is conditioned upon the relative position of the multiple screens and whether the data being displayed originated from a single-screen application or a multi-screen application.

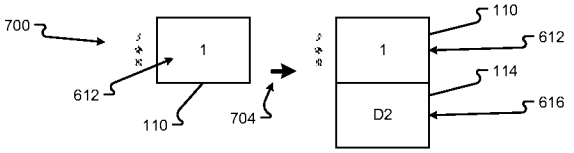


Fig. 7A

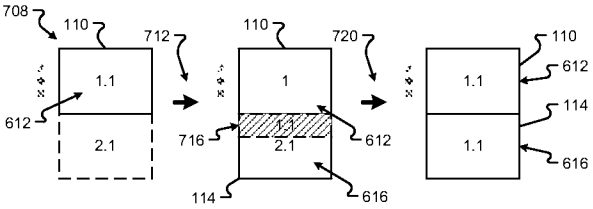


Fig. 7B

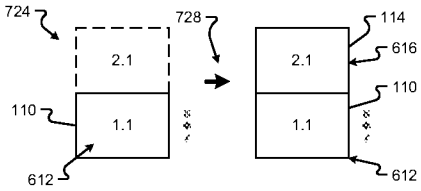


Fig. 7C

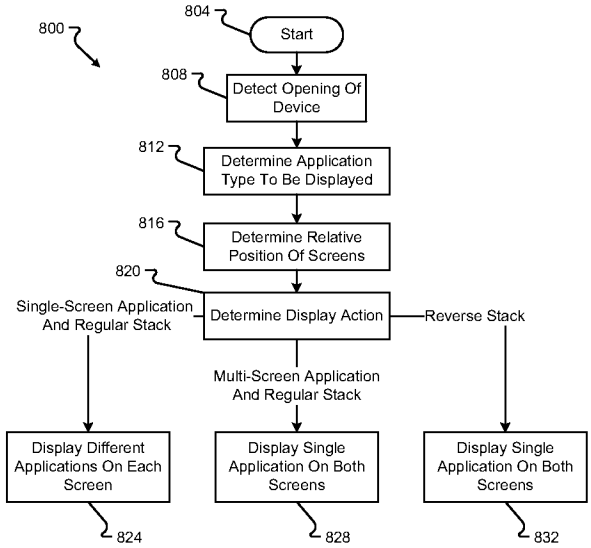


Fig. 8

APPLICATION DISPLAY TRANSITIONS BETWEEN SINGLE AND MULTIPLE DISPLAYS

SUMMARY

~~[0001]~~ There is a need for a multi-display handheld computing device that provides for enhanced power and/or versatility compared to conventional single display handheld computing devices. In particular, the present disclosure is directed to a multi-screen user device and methods for controlling data displayed thereon. The data displayed by the multiple screens of the device is dependent on the physical orientation of the device, whether the content for an application is displayed across a plurality of the multiple screens, and whether the data being displayed for the application originated from a single-screen application or a multi-screen application. More particularly, an application displayed on only a single screen or display of the device may automatically be extended across two of the screens or displays of the device when, e.g., the two screens or displays are not folded together (i.e., unfolded from one another), and a physically higher of the screens supports such a multi-screen display of its content. Such an automatic extending across two screens or displays is referred to herein as "gravity drop". Additionally, an application, whose content is displayed across two screens concurrently, can automatically be displayed, under certain circumstances, on only one of the two screens. Such a reduction from two screens or displays to only a single such screen or display is referred to herein as "reverse gravity drop". These and other needs are addressed by the various aspects, embodiments, and/or configurations of the present disclosure. Also, while the disclosure is presented in terms of exemplary embodiments, it should be appreciated that individual aspects of the disclosure can be separately claimed.

[0002] In one embodiment, the present disclosure describes a method of controlling data displayed by a multi-screen device, comprising the steps (a) through (c) following:

(a) determining that the multi-screen device has moved from a first physical orientation to a different second physical orientation, wherein the first and second physical orientations differ by the first physical orientation being one of: folded or in a landscape orientation;

(b) determining whether a first screen and a second screen of the multi-screen device are each displaying content for a same application;

(c) changing a display of the application to conform to a predetermined display configuration of the of the application on at least one of the first and second screens, wherein the display configuration is dependent upon the second physical orientation and a result of the step of determining;

wherein the above step of changing includes a step of modifying a display of the application: (a) from being displayed on only the first screen to being displayed on both the first and second screens at a same time, or (b) from being displayed on both the first and second screens at a same time to being displayed on only the first screen.

[0003] In a related embodiment, the present disclosure describes the multi-display device as having the following elements (a) through (f):

(a) a first screen including a first display area;

(b) a second screen including a second display area;

(c) an attachment configured to foldably connect the first screen with the second screen;

(d) a first sensor configured to produce a first signal indicative of a position of the hinge;

(e) a second sensor configured to produce a second signal indicative of relative positions of the first screen to the second screen; and

(f) a computer-readable medium having instructions stored thereon that include:

(f-1) a first set of the instructions configured to determine that the multi-screen device has moved from a first physical orientation to a different second physical orientation, wherein the first and second physical orientations differ by the first physical orientation being one of: folded or in a landscape orientation;

(f-2) a second set of the instructions configured to determine, whether a first screen and a second screen of the multi-screen device are each displaying content for a same application; and

(f-3) a third set of the instructions configured to change a display of the application to conform to a predetermined display configuration of the of the application on at least one of the first and second screens, wherein the

display configuration is dependent upon the second physical orientation and a result of the step of determining; and

wherein the third set of the instructions are configured to modify a display of the application: (a) from being displayed on only the first screen to being displayed on both the first and second screens at a same time, or (b) from being displayed on both the first and second screens at a same time to being displayed on only the first screen.

[0004]

[0005] The present disclosure can provide a number of advantages depending on the particular aspect, embodiment, and/or configuration. In particular, the gravity drop and reverse gravity drop operations available on the multi-display device provide an more intuitive and enhanced user experience for a user interacting with an embodiment of the multi-display device.

BRIEF DESCRIPTION OF THE DRAWINGS

[0006] Fig. 7A depicts a first sequence of gestures to invoke a first display action in accordance with embodiments of the present disclosure;

[0007] Fig. 7B depicts a second sequence of gestures to invoke a second display action in accordance with embodiments of the present disclosure;

[0008] Fig. 7C depicts a third sequence of gestures to invoke a third display action in accordance with embodiments of the present disclosure;

[0009] FIG. 8 is a flowchart of an embodiment of high level steps for performing gravity drop; and

[0010] FIG. 9 is a flow chart of an embodiment of high level steps for performing reverse gravity drop;

DETAILED DESCRIPTION

[0011] **Display Controls.**

[0012] Figs. 7A-C depict three examples of gesture sequences that can cause the device 100 to alter the display of data thereon. Specifically, Fig. 7A depicts a first gesture sequence 700 which results in a first display action. Although the first display action is depicted as being performed on the touch sensitive displays 110, 114, it should be appreciated that the first display action can alternatively, or additionally, be performed on other parts of the primary screen 104 and/or secondary screen 108. The first gesture sequence 700 begins when the device 100 is in a closed

landscape state 340. While in this state, the device 100 is displaying data in a first landscape configuration 612. The user of the device 100 then provides a first user input 704 (of the first gesture sequence 700) causing the device 100 to change from the closed landscape state 340 to an open landscape state 348. This first user input 704 may be detected by one or more of the hardware components of the device 100 as discussed herein.

[0013] In the example of Fig. 7A, the data displayed in the first landscape configuration 612 may be derived from a single-screen application, A1, which means that the application data currently being displayed on the touch sensitive display 110 is originating from the application A1 which is configured to only display data on a single screen 104 or 108. Accordingly, either before or after the MDM module 524 registers the first user input 704, the MDM module 524 determines whether the application A1 being displayed in the first landscape configuration 612 corresponds to a single-screen application. Assuming that the application A1 being displayed on the touch sensitive display 110 corresponds to a single-screen application, the MDM module 524 may be configured to select a second application or desktop D2, and cause data therefrom to be displayed in a second landscape configuration 616 of the touch sensitive display 114. Thus, in response to the first user input 704, the MDM module 524 instructs the window management module 532 to display data for the second application or desktop (D2), different from the first application (A1) being displayed on the touch sensitive display 110, on the touch sensitive display 114.

[0014] In some embodiments, the second application or desktop (D2) is selected by the window management module 532 and may correspond to a next application or desktop behind the first application in the window stack 550. More specifically, as part of selecting the second application or desktop (D2), the MDM module 524 and/or window management module 532 may reference the window stack 550 that identifies active and non-active applications or desktops, as well as display priorities for such applications or desktops, and then select the second application or desktop (D2) as the next application or desktop in the window stack 550. If no application (as opposed to desktop) is "next" behind the first application (A1), then the window management module 532 may cause either a desktop or some other default display element to be displayed on the touch sensitive display 114 as D2.

[0015] Fig. 7B depicts a second gesture sequence 708, which results in a second display action. Similar to the first display action, the second display action can be performed on any

visualization or output component of the device 100. The second gesture sequence 708 begins similarly to the first gesture sequence 700 with the device 100 being in a closed landscape state 340. The difference between the two sequences, however, is that the application (A1.1) being displayed on the touch sensitive display 110 is a multi-screen application. While the device 100 is in the closed landscape state 340, first data from the application A1.1 is displayed via the touch sensitive display 110. As indicated by the dotted lines under touch sensitive display 110 and labeling, the other touch sensitive display 114 is not actively displaying any data.

[0016] The second gesture sequence 708 continues when a first user input 712 causing the device 100 to change from the closed landscape state 340 to an open landscape state 348. This first user input 712 may be similar or identical to the first user input 704.

[0017] Still referring to Fig. 7B, before or afrer the MDM module 524 registers the first user input 712, the MDM module 524 determines that the application being displayed in the first landscape configuration 612 corresponds to a multi-screen application. Thus, in response to the first user input 712, the MDM module 524 instructs the window management module 532 to begin dropping display data from the first application A1.1 across the touch sensitive display 114 in a transition area 716. In some embodiments, the transition area 716 moves across the touch sensitive display 114 (e.g., in a downward motion that progressively fills more and more of the touch sensitive display 114 with data from the first application A1.1). The motion of the transition area 716 may be configured to emulate a dropping of the first landscape configuration 612 across the touch sensitive display 114 under the force of gravity.

[0018] Afrer the transition area 716 has completely moved across the touch sensitive display 114, the MDM module 524 may register a second event 720 that signifies completion of the gravity-based drop of the first application A1.1 from the touch sensitive display 110. At this point, data from the first application A1.1 may be displayed by both touch sensitive displays 110, 114 in any number of display configurations. In some embodiments, the application data from the first application A1.1 is automatically maximized. This may cause the first application A1.1 to either be simultaneously displayed on the touch sensitive screens 110, 114 in separate landscape configurations 612, 616 (e.g., an LD output configuration) or in the Lmax configuration 628. The application itself of the MDM module 524 may control the manner in which the first application A1.1 is displayed and whether the application data is displayed in an LD output configuration or an Lmax configuration 628.

[0019] In one embodiment, the type of data displayed from the first application A1.1 on the touch sensitive displays 110, 114 may vary depending upon the nature of the first application. As a non-limiting example, the type of user input available for interfacing with the first application A1.1 may change from a first type of user input (e.g., a writing-capture area, a half QWERTY keyboard) to a second type of user input (e.g., a full QWERTY keyboard or combination writing-capture area and half QWERTY keyboard) when the first application A1.1 is automatically maximized across the touch sensitive displays 110, 114 during the gravity drop. Likewise, the type of user output may vary from a first type of user output to a second type of user output when the first application A1.1 is automatically maximized across the touch sensitive displays 110, 114 during the gravity drop.

[0020] Fig. 7C depicts a third gesture sequence 724 for obtaining a resulting third display action, wherein the device 100 is in a "reverse landscape" state 340a where the configurable area 112 is on the opposite side (i.e., right side) of the device when the display 110 faces the user. Similar to the first and second display actions, the third display action can be performed on any visualization or output component of the device 100. The third gesture sequence 724 begins similarly to the other gesture sequences 700, 708 with the device 100 being in a closed landscape state; i.e., the reverse landscape state 340a. While the device 100 is in the closed reverse landscape state 340a, a first set of data from the first application A1.1 is displayed via the touch sensitive display 110. As indicated by the dotted lines above the touch sensitive display 110 and labeling, the other touch sensitive display 114 is not actively displaying any data.

[0021] The third gesture sequence 724 has a first user input 728 causing the device 100 to change from the closed reverse landscape state 340a to an open reverse landscape state 348a. This first user input 728 may be similar or identical to the other user inputs 704, 712, except that the secondary screen 108 is opened above the primary screen 104. Also, after the first user input 728 is received, the data from the first application A1.1 is already being displayed on the bottom touch sensitive display 110 (e.g., the display 110 which is positioned below the other touch sensitive display 114). The relative positions of the displays 110, 114 (or more specifically the screens 104, 108) may be determined from data provided by the accelerometer 176 or some other appropriate hardware component of the device 100.

[0022] In some embodiments, when the touch sensitive display 114 is opened above the active touch sensitive display 110 as in Fig. 7C, the application (A1.1) in view on the touch sensitive

display 110 maintains its original display configuration. Specifically, the first application A1.1 being displayed on the touch sensitive display 110 may continue to be displayed in a first landscape configuration 612 while data from a second application A2 may be displayed in a second landscape configuration 616 via the other touch sensitive display 114 since no gravity drops occurs to pull the display of the first application A1.1 onto screen 114 which is thereabove.

[0023] The display actions performed on the touch sensitive displays 110, 114 may be controlled by the collective actions of the MDM module 524 and window management module 532 based on one or more event interrupts received from various hardware devices. Such interrupts may be registered and interpreted by the task management module 540. In selecting the data that is to be displayed via the touch sensitive display 114, the MDM module 524 and/or window management module 532 may identify the next application in a display stack as discussed above. If no application is "next" behind the first application A1.1, then the MDM module 524 and/or window management module 532 may cause either the desktop (e.g., a display rendered for the OS) or some other default display element to be displayed on the touch sensitive display 114.

[0024] Although the three examples of display actions were given in instances where an application was initially displayed on touch sensitive display 110, it should be appreciated that if the touch sensitive display 114 is initially displaying application data then the subsequent processing steps and display actions may be reversed. Specifically, if data from, e.g., the application (A1 or A1.1) is initially displayed via touch sensitive display 114 when device 100 is in a closed landscape state 340, then after the first user input 704, 712, 728 is received, the window management module 532 may manipulate the subsequent display of data on the touch sensitive display 110 depending upon the position of the touch sensitive display 110 relative to the touch sensitive display 114 and/or whether the application having data displayed is a single-screen application or a multi-screen application.

[0025] Referring now to Fig. 8, this figure presents a high level flowchart of the steps performed by the device 100 in determining whether a gravity drop is to be performed as described hereinabove. In particular, the steps of Fig. 8 are, in one embodiment, performed by the MDM module 524. Accordingly, in step 804, a determination is made as to whether the device 100 is oriented in (a) the open landscape state 348, or (b) the open reverse landscape state 348a, or (c) another orientation. If it is determined that the device 100 is oriented in neither the

open landscape state 348 nor the open reverse landscape state 348a, then any further gravity drop processing is terminated (e.g., at least until the device 100 is reoriented, or another application is considered for gravity drop). Thus, the flowchart terminates at terminator 808 wherein the device 100 (more particularly, the MDM module 524) ceases the current gravity drop processing. If, however, it is determined in step 804 that the device 100 is in the open landscape state 348 or the open reverse landscape state 348a, then step 812 is performed, wherein further determinations are made to assure that the device 100 is properly configured for performing a gravity drop (e.g., from display 110 to display 114 as shown in Fig. 7B). In particular, step 812 determines whether the following additional three conditions are satisfied:

(a) In the current device orientation, is the display 110 or 114 that is physically uppermost (e.g., physically higher) displaying an application in a single screen mode (i.e., displaying on only one of the displays 110 or 114)?

(b) Can this application be displayed (i.e., does it support being displayed) on more than one display (e.g., both displays 110 and 114)?

(c) Has gravity drop for this application not been entirely disabled? Note that for certain applications, gravity drop should never occur. For example, in a camera or video application where the application is only operable for taking photos or videos when the device 100 is in a folded configuration, or, a game application where the game is played by two players, each viewing only one of the displays 110 or 114, and wherein the device is in the orientation as shown in, e.g., Fig. 1I.

[0026] If one or more of the conditions (a) through (c) immediately above are not satisfied, then again any further gravity drop processing is terminated (at least until the device 100 is reoriented, or another application is considered for gravity drop). Thus, the flowchart terminates at terminator 808 wherein the device 100 (more particularly, the MDM module 524) ceases the current gravity drop processing.

[0027] However, if all of the conditions (a) through (c) immediately above are satisfied, then step 816 is performed where further conditions are evaluated for determining whether gravity drop should occur. More specifically, the present step determines whether the current settings for the application would prevent a gravity drop from occurring (e.g., since such a gravity drop would not make sense to the user, or would decrease a user's experience with the device 100). For example, if the user has input a request that another application, on the lowermost one of the

displays 110 or 114, remain visible on this lowermost display, then the determination in step 816 would be positive. As a further example, if at least some applications may only have gravity drop performed thereon when the lowermost display is not actively outputting new information to the user such as if a user is watching a movie or video. Accordingly, if the determination in step is positive, then in step 820, one or more gravity drop parameters may be set to indicate that no gravity drop is to be performed on the application's display as currently configured. Alternatively, if the determination in step 816 is negative, then in step 824, gravity drop is performed on the application.

[0028] Note that machine instructions corresponding to the flowchart of Fig. 8 may be performed by the device 100 when: (a) the device enters one of the open landscape configurations 348 or 348a (e.g., by the user unfolding the device, and/or rotating the device into a landscape state), or (b) a user explicitly requests the device to perform a gravity drop on a displayed application.

[0029] For an application on which a gravity drop is performed, there are of course conditions when the application's display should preferably revert back to a single screen display. In particular, such a reversion may configure the application's display to be on the same one of the displays 110 or 114 from which gravity drop occurred. The activation of a procedure to perform such a reversion is referred to herein as "reverse gravity drop". A reverse gravity drop may be performed: (a) substantially automatically, e.g., when the MDM module 524 receives appropriate input for directing the reverse gravity drop to occur, or (b) the user may explicitly request a reverse gravity drop on a displayed application upon which a gravity drop was performed.

[0030] In Fig. 9 a high level flowchart is shown providing the steps performed, e.g., by the MDM module 524 for performing a reverse gravity drop. In step 904, a determination is made as to whether the device 100 is being oriented from one of the landscape states 348 or 348a to the dual-portrait orientation. If the result of decision step 904 is negative, then any further current processing regarding performing a reverse gravity drop is terminated (e.g., at least until the device 100 is reoriented, or another application is considered for gravity drop). Thus, the flowchart terminates at terminator 908 wherein the device 100 (more particularly, the MDM module 524) ceases the current reverse gravity drop processing. If, however, there result of step 904 is positive, then step 912 is performed, wherein further determinations are made to assure that the device 100 is properly configured for performing a reverse gravity drop. In particular,

step 912 determines whether: (a) the target application (for a reverse gravity drop) is indeed displayed on both displays 110 and 114, and (b) there is configuration data identifying that a reverse gravity drop is pending for being performed on the target application when specific conditions are satisfied. Note that such "reverse gravity drop pending" configuration data is created for a target application only whenever a gravity drop has previously performed on the target application, and deleted whenever a reverse gravity drop is performed on the target application. Thus, if there is "reverse gravity drop pending" configuration data for a target application, this is intended to mean that a previous gravity drop for the target application occurred, and the target application is currently displayed according to this gravity drop.

[0031] If the result to the decision step 912 is negative, then terminator 908 is encountered and processing for a reverse gravity drop ceases. However, if the result to the decision step 912 is positive, then an additional set of conditions is checked for determining if a reverse gravity drop can be performed on the display of the target application. In particular, in step 916, the following conditions (a) and (b) are evaluated by the MDM module 524:

(a) Is the device 100 being folded?

(b) Is the device 100 being docked or undocked?

That is, if the MDM module 524 receives input indicating that either of these conditions is satisfied, then in step 920, the reverse gravity pending configuration data for the target application is at least confirmed to be deleted (or at least rendered invalid). In particular, if the device 100 is being folded, then in one embodiment, any reverse gravity drop pending configuration data is automatically deleted since for various applications, a folding of the device 100 is permissible (and desirable) when the application is displayed on both the displays 110 and 114 (e.g., certain game applications). Accordingly, whether a dual screen display of an application (e.g., due to a previous gravity drop operation) should revert back to being displayed on only one of the displays 110 or 114 may be dependent on the particular application requesting or not requesting a reverse gravity drop.

[0032] Alternatively, if it is determined in step 916 that device 100 is not being folded, and not being docked, and not being undocked, then the target application remains eligible for a reverse gravity drop to be performed thereon. Accordingly, regardless of the outcome of step 916, decision step 924 is performed to evaluate whether a reverse gravity drop is actually to be performed on the display of the target application. In particular, when step 924 determines that

when one or more of the following conditions (a) through (c) are satisfied, a reverse gravity drop is to be performed in step 928: (a) data for a reverse gravity drop pending is still valid, or (b) the user provides an input to explicitly request the target application to be displayed on only a single one of the displays 110 or 114, or (c) the target application requests a reverse gravity drop. Accordingly, depending on the outcome from decision step 924, reverse gravity drop is performed (in step 928), or processing for such a reverse gravity drop terminates without a reverse gravity drop being performed.

What is claimed is:

1. A method of controlling data displayed by a multi-screen device, comprising:

determining that the multi-screen device has moved from a first physical orientation to a different second physical orientation, wherein the first and second physical orientations differ by the first physical orientation being one of: folded or in a landscape orientation;

determining whether a first screen and a second screen of the multi-screen device are each displaying content for a same application;

changing a display of the application to conform to a predetermined display configuration of the of the application on at least one of the first and second screens, wherein the display configuration is dependent upon the second physical orientation and a result of the step of determining;

wherein the step of changing includes a step of modifying a display of the application: (a) from being displayed on only the first screen to being displayed on both the first and second screens at a same time, or (b) from being displayed on both the first and second screens at a same time to being displayed on only the first screen.

2. The method of claim 1, wherein the first screen is determined to be above the second screen.

3. The method of claim 2, wherein a relative position of the first to the second screen is determined, at least in part, based on data received from an accelerometer.

4. The method of claim 2, further comprising:

determining whether the application is a single-screen application or a multi-screen application; and

wherein controlling data displayed on the second screen in an open landscape state is also based on whether the application is a single-screen application or a multi-screen application.

5. The method of claim 4, wherein the application is a multi-screen application, the method further comprising:

automatically causing data from the application to be displayed on the second screen in a landscape state such that the device is not in a folded closed configuration.

6. The method of claim 4, wherein the application is a single-screen application and wherein data from a different second application is concurrently displayed on the second screen.

7. The method of claim 1, wherein: (a) the second physical orientation is a portrait orientation, (b) the first physical orientation is a landscape orientation, (c) prior to the step of changing, the first and second screens each display content for the application, and the step of modifying includes displaying the application on only the first screen.

8. The method of claim 7, wherein data from a second application is displayed on the second screen simultaneous with data from the application being displayed on the first screen.

9. The method of claim 1, wherein: (a) the second physical orientation is an unfolded orientation of the device, (b) the first physical orientation is a folded orientation, (c) prior to the step of changing, the only first screen displays the content of the application, and the step of modifying includes displaying concurrently at least a portion of the content on each the first and second screens.

10. A non-transitory computer-readable medium having machine instructions stored thereon, the instructions comprising:

a first set of the instructions configured to determine that a multi-screen device has moved from a first physical orientation to a different second physical orientation, wherein the first and second physical orientations differ by the first physical orientation being one of: folded or in a landscape orientation;

a second set of the instructions configured to determine whether a first screen and a second screen of the multi-screen device are each displaying content for a same application; and

a third set of the instructions configured to change a display of the application to conform to a predetermined display configuration of the of the application on at least one of the first and second screens, wherein the display configuration is dependent upon the second physical orientation and a result of the step of determining;

wherein the third set of the instructions are configured to modify a display of the application: (a) from being displayed on only the first screen to being displayed on both the first

and second screens at a same time, or (b) from being displayed on both the first and second screens at a same time to being displayed on only the first screen.

11. The computer-readable medium of claim 10, wherein the first screen corresponds to a primary screen and wherein the second screen corresponds to a secondary screen.

12. The method of claim 10, wherein the first screen is determined to be above the second screen, and wherein data displayed on the second display in an open landscape state is also based on whether the application is a single-screen application or a multi-screen application.

13. The computer-readable medium of claim 12, wherein the application is a multi-screen application and wherein the instructions configured to control data displayed on the first and second displays also generates a transition area on the second display that moves across and progressively fills the second display with data from the first application.

14. The computer-readable medium of claim 13, wherein the data from the application is automatically displayed on both the first and second displays when: (a) the second physical orientation is an unfolded orientation of the device, (b) the first physical orientation is a folded orientation, (c) prior to a corresponding execution of the instructions for modifying the display of the application, the only first screen displays the content of the application, and afrer the execution of the instruction for modifying, concurrently displaying at least a portion of the content on each the first and second screens. .

15. The computer-readable medium of claim 12, wherein the application is a single-screen application and wherein data from a second application is automatically displayed on the second screen immediately afrer the device transitions to the open landscape state.

16. A multi-screen user device, comprising:

a first screen including a first display area;

a second screen including a second display area;

an attachment configured to foldably connect the first screen with the second screen;

a first sensor configured to produce a first signal indicative of a position of the hinge;

a second sensor configured to produce a second signal indicative of relative positions of the first screen to the second screen; and

a computer-readable medium having instructions stored thereon that include:

a first set of the instructions configured to determine that the multi-screen device has moved from a first physical orientation to a different second physical orientation, wherein the first and second physical orientations differ by the first physical orientation being one of: folded or in a landscape orientation;

a second set of the instructions configured to determine, whether a first screen and a second screen of the multi-screen device are each displaying content for a same application; and

a third set of the instructions configured to change a display of the application to conform to a predetermined display configuration of the of the application on at least one of the first and second screens, wherein the display configuration is dependent upon the second physical orientation and a result of the step of determining; and

wherein the third set of the instructions are configured to modify a display of the application: (a) from being displayed on only the first screen to being displayed on both the first and second screens at a same time, or (b) from being displayed on both the first and second screens at a same time to being displayed on only the first screen.

17. The device of claim 16, wherein the application is a multi-screen application and wherein the third set of instructions is further configured to generate a transition area on the second display that moves across and progressively fills the second display with data from the application.

18. The device of claim 17, wherein the data from the application is automatically displayed on both the first and second displays when: (a) the second physical orientation is an unfolded orientation of the device, (b) the first physical orientation is a folded orientation, (c) prior to a corresponding execution of the instructions for modifying the display of the application, the only first screen displays the content of the application, and after the execution of the instruction for modifying, concurrently displaying at least a portion of the content on each the first and second screens..

19. The device of claim 17, wherein the data from the application is automatically displayed on only the first screen when (a) the second physical orientation is a portrait orientation, (b) the first physical orientation is a landscape orientation, (c) prior to a

corresponding execution of the third set of instructions, the first and second screens each display content for the application, and afrer corresponding execution of the third set of instructions, displaying the application on only the first screen.

20. The device of claim 17, wherein the first screen is determined to be above the second screen, based at least partially on the second signal.

ABSTRACT

A multi-screen user device and methods for controlling data displayed are disclosed. The data displayed by the multiple screens of the device is dependent on the physical orientation of the device, whether the content for an application is displayed across a plurality of the multiple screens, and whether the data being displayed for the application originated from a single-screen application or a multi-screen application.

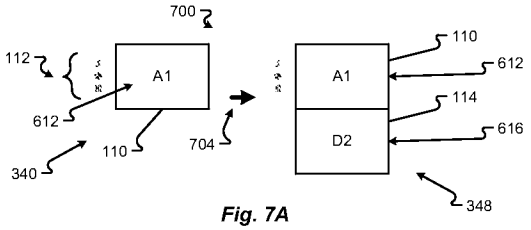


Fig. 7A

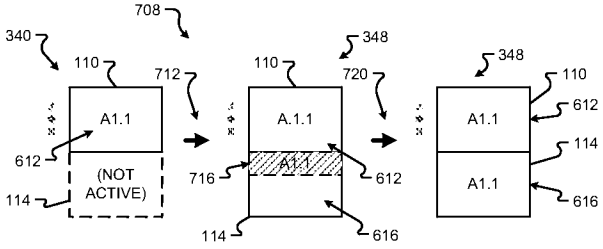


Fig. 7B

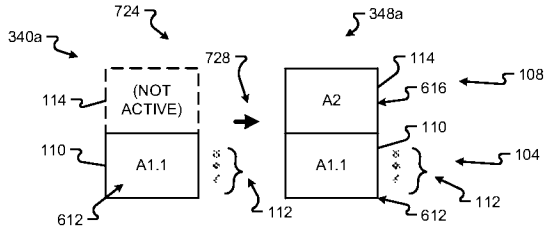


Fig. 7C

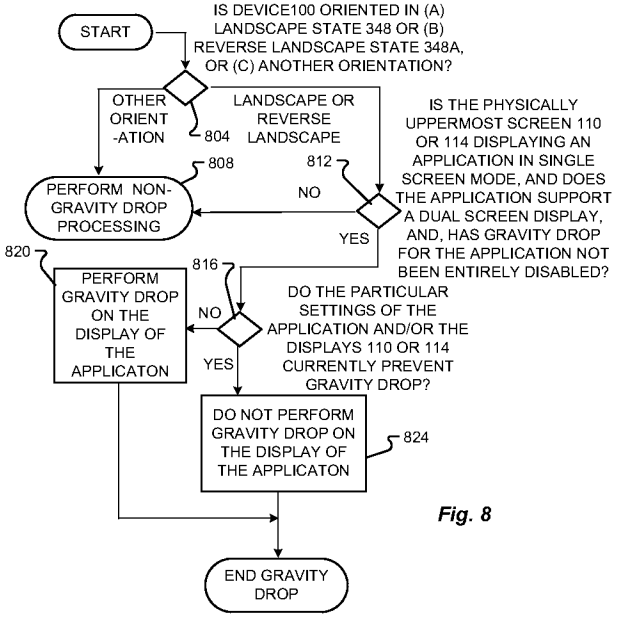


Fig. 8

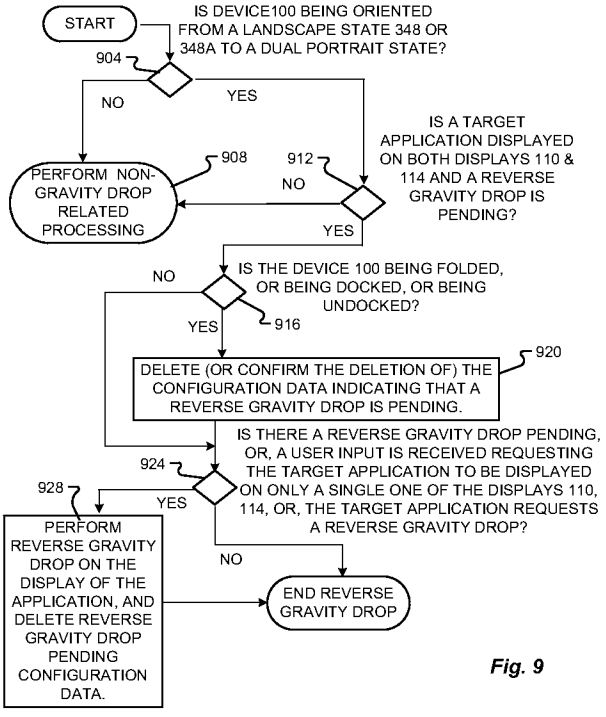


Fig. 9

GESTURE CAPTURE FOR MANIPULATION OF PRESENTATIONS ON ONE OR MORE DEVICE DISPLAYS

SUMMARY

[0001] There is a need for a dual multi-display handheld computing device that provides for enhanced power and/or versatility compared to conventional single display handheld computing devices. These and other needs are addressed by the various aspects, embodiments, and/or configurations of the present disclosure. Also, while the disclosure is presented in terms of exemplary embodiments, it should be appreciated that individual aspects of the disclosure can be separately claimed. More particularly, there is a need for an intuitive technique for inputting user gestures into a handheld computing device that allows a user to more easily manipulate different types of screen display presentations, such as desktop presentations as opposed to application window presentations (the terms "desktop" and "application window" being described and distinguished hereinbelow). Since applications may be downloaded from a wide variety of sources and may respond to an application specific set of user gestures, user gesture inputs are neither uniform nor consistent between application windows. Furthermore, since device application windows typically require the user to provide application specific user gesture inputs directly to the displayed application windows, such application specific user inputs prevent the use of a uniform or consistent set of user gesture inputs directly to displayed application windows for performing, e.g., window manipulation tasks such as minimizing, maximizing, and moving application windows from one screen display to another, as well as rearranging in a stack of application windows, activating a display of an application window across multiple screen displays, and/or reducing a display of an application window to a single screen display that is currently displayed across multiple screen displays. Accordingly, user gesture inputs for manipulating application windows according to a consistent or uniform set of window based user gesture inputs has heretofore been problematic. On-the-other-hand, for displays of desktops, since the device typically has control over the operation and manipulation of desktop displays, user gesture inputs for such desktop operation and manipulation can be made or forced to be uniform. Accordingly, the present disclosure describes a handheld device which retains the uniformity that a device typically can impose on the manipulation of desktop

displays while also imposing a similar uniformity and consistency on the manipulation of application windows. In one embodiment, the present disclosure is directed to performing the following steps (a) through (h) for configuring a device having at least one display screen for displaying one or more of screen display presentations and for displaying one or more operably different screen display presentations, wherein the screen display presentations are able to be manipulated via user gesture input differently from the operably different screen display presentations:

(a) receiving a first user gesture input to a gesture capture area associated with the at least one screen, the gesture capture area being separate from the at least one display screen;

(b) determining whether a first of the one or more of screen display presentations is displayed on the at least one screen;

(c) when the first screen display presentation is determined to be displayed, perform a step of interpreting the gesture input for determining a predetermined operation to apply to the first screen display presentation;

(d) when none of the one or more screen display presentations are determined to be displayed in the step of determining, perform a step of preventing the device from using the gesture input to change the at least one display screen;

(e) receiving a second user gesture input in response to a second user gesture input that includes user contact input directly to the at least one screen;

(f) second determining whether one of the operably different screen display presentations is displayed on the at least one screen;

(g) when the one operably different screen display presentation is determined to be displayed on the at least one screen, performing a step of second interpreting the second gesture input for determining a predetermined second operation to apply to the one operably different screen display presentation; and

(h) when none of the operably different screen display presentations are determined in the step of second determining, perform a step of second preventing the device from using the second gesture input to change the at least one display screen.

[0002] Moreover, is it an aspect of the present disclosure that there is a predetermined collection of a plurality of user gesture input descriptions for recognizing user gesture inputs that are interpretable in the step of interpreting in (c) above, wherein:

(i) the one or more screen display presentations includes a plurality of application windows,

(ii) each of a plurality of software applications provide a corresponding one of the plurality of application windows,

(iii) each of the user gesture input descriptions (UGI) of the predetermined collection is operable for recognizing user gesture inputs for each application window (AW) of the plurality of application windows when: (1) the user gesture input UGI is an instance of the first user gesture input, and (2) the application window AW is an instance of the first screen display presentation for which the user gesture input UGI is an input.

[0003] Note that each of the software applications of the plurality of software applications in (ii) immediately above may be downloaded (via a communications network such as the Internet) to the handheld device from a software vendor that is independent other software vendors from which other of the plurality of software applications can be downloaded.

[0004] The present disclosure can provide a number of advantages depending on the particular aspect, embodiment, and/or configuration. In particular, the present disclosure simplifies a user's experience in providing gesture input to the device. More particularly, for each of the plurality of device displays, the present disclosure provides a corresponding separate area associated therewith for receiving user gesture input, wherein this gesture input (to the separate area) is interpreted as applying to application windows and not to desktop displays. Thus, since user gesture input for manipulating desktops can be input directly to a display presenting the desktop, a user can readily specify that how a displayed application window is to be manipulated by inputting user gestures into the corresponding separate area associated with the display presenting the application window.

BRIEF DESCRIPTION OF THE DRAWINGS

[0005] FIG. 10A illustrates processing performed by an embodiment of the Gesture Module 576, wherein a combination flick-drag gesture is applied to the gesture capture area 124, and the Gesture Module 576 ignores the gesture since there are no application windows in the displays 110 and 114 to be operated on according to the gesture;

[0006] FIG. 10B illustrates processing performed by the device 100, wherein a combination flick-drag gesture is applied directly to the desktop of display 114, the desktops are shifted to the left;

[0007] FIG. 10C illustrates processing performed by an embodiment of the Gesture Module 576, wherein a combination flick-drag gesture is applied to the gesture capture area 124, and the Gesture Module 576 since there is an application window in the display 110 to which the gesture can be applied, the Gesture Module 576 shifts the application A1 to the right;

[0008] FIG. 10D is similar to Fig. 10C except that in the window stack for the display 110, there is another application window A3 in the stack which is then displayed in the display 110 when the application window A1 is shifted to the right;

[0009] 10E illustrates processing performed by an embodiment of the Gesture Module 576, wherein a combination flick-drag gesture is applied to the display 110, and the Gesture Module 576 ignores the gesture since no desktop is displayed in the display 110 to be operated on according to the gesture;

[0010] FIG. 11 is a high level flowchart of the processing performed by the Gesture Module 576 for providing the display changes shown in Figs. 10A through 10D;

DETAILED DESCRIPTION

[0100] In at least one embodiment of the device 100, the MDM 524 (and its Gesture Module 576) may be operable for interpreting gestures in the gesture capture areas 120 and 124 so that they may be used to manipulate application windows, e.g., to minimize, maximize, and move application windows between the displays 110 and 114, as well as to rearrange the application windows in the window stack, to active an application window across multiple screen displays, and/or reduce an application window to a single screen display that is currently displayed across displays 110 and 144. However, in various embodiments, user gesture input to the gesture capture areas 120 and 124 is not used to manipulate displayed desktops. Referring to Fig. 10A, assuming the flick gesture 404 and the drag gesture 400 are concurrently performed in, e.g., the gesture capture area 124, wherein each of the displays 110 and 114 display a desktop (respectively, D1 and D2), then as shown in the right-hand portion of this figure, such gestures do not affect the displays 110 and 114. However, referring to Fig. 10B, the same combination of gestures (e.g., flick gesture 404 and drag gesture 400) provided on the display 114 cause the displayed desktops D1 and D2 to shift to the left. More particularly, in Fig. 10B, the flick-drag

combination gesture on display 114 shifts the application D1 off the display 110, causes the application D2 to shift from the display 114 to the display D1, and for another desktop, D3, this later desktop shifts onto the display 114. Accordingly, depending on the window stack implementation (e.g., the implementation of Figs. 7A and 7B, or that of Figs. 7C-7E), the window stack(s) may be changed to reflect the change in which desktops are displayed to the user. Thus, for the embodiment of the window stacks as provided in Fig. 7A, the desktop D2 changes window stack while desktops D1 and D3 change positions within their respective window stacks, and of course, the logical data structures 800 for each of the desktops D1, D2, and D3 changes accordingly.

[0101] Referring to Fig. 10C, this figure illustrates the same flick-drag combination gesture used in the gesture capture area 120 for moving the application window A1 for an application (not a desktop) from the display 110 to the display 114. Note that in performing such moving of the application window A1, the respective window stacks (assuming a window stack embodiment as per Fig. 7A) for the displays 110 and 114 are changed. In particular, when the application window A1 moves from the display 110 (due to the flick-drag combination gesture in area 120), this window also moves from the top of the window stack for the display 110 to the top of the window stack for the display 114, and is removed from the window stack for the display 110. Thus the movement of application windows between displays 110 and 114 will change their corresponding window stacks. Moreover, as shown in Fig. 10C, assuming the application window A1 were the only application window in the window stack for the display 110, then if a desktop D1 for the display 110 is on the window stack for display 110 immediately below application window A1, then desktop D1 is presented on the display 110. However, as shown in Fig. 10D, if there is an application window A3 in the window stack immediately below application window A1 in the window stack for display 110, then upon activation of the flick-drag combination gesture in area 120, application window A3 is displayed in display 110.

[0102] In a further embodiment shown in Fig. 10E, this figure illustrates that the same flick-drag gesture combination applied directly to a display 110 or 104, wherein an application window (labeled A1 in Fig. 10E) is displayed will yield no change to the displays 110 and 114.

[0103] Note that in addition to the combination flick-drag gesture used illustratively above, other gestures and combination gestures input to the gesture capture areas 120 and 124. In

particular, the following gestures and their interpretations (as applied to application windows) are contemplated herein:

[0104]

Gesture (or combination gesture)	Interpretation by the Gesture Module 576
Drag 400	Depending on what application window is active (e.g., displayed on the corresponding display 110 or 124 and has focus) to be manipulated by the drag 400 gesture, the application window may be, e.g., moved, maximized or minimized. For example, if such an application window is displayed but minimized (e.g., as an icon), then a drag may maximize this window. Alternatively, a drag gesture for such an application window that is maximized may minimize the application window.
Long Drag	A long-drag gesture is a drag gesture on one of the gesture capture areas 120, 124. A long drag is like a regular drag gesture 400, except longer. In practice what this means is that one is performing several individual drag operations in sequence, as long as the long-drag is being performed. When a regular drag 400 operation is performed, it moves applications, or minimizes or maximizes dual-screen applications. A long-drag gesture will do the exact same things, except several of them in sequence automatically for as long as the drag is performed. Essentially, a long-drag gesture equals several individual drag operations performed sequentially. Therefore, by using a long-drag, it is possible to move several single and/or dual-screen applications, depending on the length of gesture. For example, if 10 applications are displayed on the display screen 114, wherein each of the applications is a single-screen application. In performing a long drag from right to left, the application in view on the display 114 (app A) starts moving towards the display 110. Upon continuing the long drag,

	eventually the application (app A) is positioned on display 110 and on the display 114, another (app B) is fully exposed. If the user stopped "long drag" at this point, only a regular drag 400 would have been performed. However, by continuing the drag gesture (i.e., a long-drag); the application on the display 114 (app B) starts moving towards the display 110. Thus, on the display 114, a third application (app C) becomes visible. Eventually, app B is on the display 110 (hiding app A), and app C is fully exposed on the display 114. In essence, two "regular" drag operations will have been performed with one "long drag" gesture. If app A had been dual-screen capable, a long-drag gesture could have first maximized it, then minimized it to the display 110.
Flick 404	Moves or minimizes the active application window similarly to a drag 400. However, no maximization of the window is applied.
Pinch 408	Swaps the position of application windows displayed on the corresponding displays 110 or 124. Ignored if the device 100 is closed. Also "ignored" for maximized dual-screen applications.
Spread 410	Displays the application manager (i.e., a "pop-up window" that displays the application windows in the window stack)
Pin and Drag combination 416 (Fig. 4H)	Holds a pinned application window in place while moving another application window in the window stack. In more detail, a pin and drag gesture allows for more advanced window navigation where one application is held in view while the window stack is navigated beneath. The pin and drag gesture is performed by holding down one finger on one of the gesture capture areas 120, 124, and performing a drag or flick gesture with another finger. Essentially, Pin and drag allows for keeping one application in view while the rest of the applications in the window stacks are navigated. It's like performing drag gestures but one of the

	applications in view is not affected by the gestures: it's being "held down" and not allowed to move while the others do. A Pin and drag gesture is used when the device 100 is open (e.g., with both displays 110, 114 being active.

[0105] The device 100 functionality described immediately above for manipulating only application windows (via user gestures input the gesture capture area 120 or 124) may be provided a particular embodiment of the MDM 524 that performs the flowchart shown in Fig. 11 and now described. Accordingly, in step 1104 of Fig. 11, the Gesture Module 576 (of the MDM 524) detects a user gesture input to one (or both) of the gesture capture areas 120 or 124. In one embodiment, such detection is performed by comparing the input user gesture with a predetermined collection of gesture template data descriptions resident on the device 100. Assuming, the Gesture Module 576 is able to identify the gesture input as one of the predetermined user gestures for the collection, this module notifies the Display Configuration Module 568 (of the MDM 524) of: (i) the gesture identity of the gesture input, and (ii) data identifying from where the gesture input as received (e.g., from one of the gesture capture areas or one of the displays 110 and 114).

[0106] The Display Configuration Module 568 determines for each gesture input to the device 100 (whether this input is to one of the displays 110, 114, or to one of the gesture capture areas 120 or 124) if the gesture identity (for the gesture input) is applicable or not for initiating a predetermined gesture corresponding action to perform on an application window or a desktop displayed on the display 110 or 114 to which the gesture input corresponds. Thus, if the gesture input is from one of the displays 110 or 114, then Display Configuration Module 568 will only notify the Window Management Module 532 to modify the display receiving the gesture input if this display displays a desktop. Alternatively, if the gesture input is from one of the gesture capture areas 120 or 124, then Display Configuration Module 568 will only notify the Window Management Module 532 to modify the display corresponding to the gesture input if this display displays an application window.

[0107] Accordingly, in step 1108, the Display Configuration Module 568 determines whether there is an application window displayed on the display 110 or 114 corresponding to the gesture capture area 120 or 124 that received the inputted user gesture. If no application window is displayed on the display 110 or 114 corresponding to the gesture capture area 120 or 124 that

received the inputted user gesture, then in step 1112, Display Configuration Module 568 prevents the user gesture input from being used to change the displays 110 and 114, and instead discards such user gesture input. However, if the display 110 or 114 (corresponding to the gesture capture area receiving the inputted user gesture) displays an application window, then the Display Configuration Module 568 uses the identity of the gesture input for determining how to change the display 110 or 114.

[0108] Note that it is also within the scope of the present disclosure that the above described different processing performed on one or more desktops as opposed to the processing performed on one or more application windows, according to user gesture input, can be reversed. For example, in an alternative embodiment, for each of the displays 110 and 114: (i) a desktop displayed on the display (110 or 114) is manipulated via gesture input to the corresponding gesture capture area 120 or 124, (ii) when an application window is displayed on the display (110 or 114), such gesture input to the corresponding gesture capture area 120 or 124 is not used for manipulating the displayed application window, (iii) an application window displayed on the display (110 or 114) is manipulated via gesture input that includes user input that directly contacts the display (110 or 114), and (iv) when a desktop is displayed on the display (110 or 114), such a gesture input that includes user input that directly contacts the display (110 or 114) is not used for manipulating the desktop.

What is claimed is:

1. A method for configuring a device having at least one display screen for displaying one or more of screen display presentations and for displaying one or more operably different screen display presentations, wherein the screen display presentations are able to be manipulated via user gesture input differently from the operably different screen display presentations, comprising:

receiving a first user gesture input to a gesture capture area associated with the at least one screen, the gesture capture area being separate from the at least one display screen;

determining whether a first of the one or more of screen display presentations is displayed on the at least one screen;

when the first screen display presentation is determined to be displayed, perform a step of interpreting the user gesture input for determining a predetermined operation to apply to the first screen display presentation;

performing the predetermined operation;

when none of the one or more screen display presentations are determined to be displayed in the step of determining, perform a step of preventing the device from using the user gesture input to change the at least one display screen;

receiving a second user gesture input in response to a user gesture input that includes user contact directly to the at least one screen;

second determining whether one of the operably different screen display presentations is displayed on the at least one screen;

when the one operably different screen display presentation is determined to be displayed on the at least one screen, performing a step of second interpreting the second user gesture input for determining a predetermined second operation to apply to the one operably different screen display presentation; and

when none of the operably different screen display presentations are determined in the step of second determining, perform a step of second preventing the device from using the second user gesture input to change the at least one display screen.

2. The method of Claim 1, wherein the gesture capture area includes a capacitive touch sensitive area for receiving the user gesture input.

3. The method of Claim 2, wherein the gesture capture area is not capable of displaying any of the screen display presentations, and is not capable of displaying any of the operably different screen display presentations.

4. The method of Claim 1, wherein the first screen display presentation includes an application window.

5. The method of Claim 1, wherein the one operably different screen display presentation includes a desktop.

6. The method of Claim 1, wherein the user gesture input includes one of: a drag, a flick, a pinch, and a spread.

7. The method of Claim 1, wherein the second user gesture input includes one of: a drag, a flick, a pinch, and a spread.

8. The method of Claim 1, wherein each of the first screen display presentation and the one operably different screen display presentation has corresponding data included in a corresponding stack, wherein a user inputs user gestures to the device rearranging entries in each stack.

9. The method of Claim 1, wherein each of the user gesture input and the second user gesture input is for a same user gesture.

10. The method of Claim 1, wherein the at least one display screen is one of a plurality of display screens of the device, and the predetermined operation moves the first screen display presentation from the at least screen to another of the plurality of screens.

11. The method of Claim 8, wherein the predetermined second operation moves the one operably different screen display presentation from the at least screen to another of the plurality of screens.

12. The method of Claim 1, wherein there is a predetermined collection of a plurality of user gesture input descriptions for recognizing user gesture inputs that are interpretable in the step of interpreting,

wherein the one or more screen display presentations includes a plurality of application windows,

wherein each of a plurality of software applications provide a corresponding one of the plurality of application windows,

wherein each of the user gesture inputs (UGI) of the predetermined collection is operable for recognizing user gesture inputs for each application window (AW) of the plurality of application windows when: (1) the user gesture input UGI is an instance of the first user gesture input, and (2) the application window AW is an instance of the first screen display presentation for which the user gesture input UGI is an input.

2.

3. 13. The method of Claim 12, wherein each of the sofrware applications is operable on the device independently of each of the other of the software applications.

4. 14. A computer readable medium for configuring a device having at least one display screen for displaying one or more of screen display presentations and for displaying one or more operably different screen display presentations, wherein the screen display presentations are able to be manipulated via user gesture input differently from the operably different screen display presentations, comprising:

5. machine instructions for performing the following steps:

receiving a first user gesture input to a gesture capture area associated with the at least one screen, the gesture capture area being separate from the at least one display screen;

determining whether a first of the one or more of screen display presentations is displayed on the at least one screen;

when the first screen display presentation is determined to be displayed, perform a step of interpreting the user gesture input for determining a predetermined operation to apply to the first screen display presentation;

performing the predetermined operation;

when none of the one or more screen display presentations are determined to be displayed in the step of determining, perform a step of preventing the device from using the user gesture input to change the at least one display screen;

receiving a second user gesture input in response to a user gesture input that includes user contact directly to the at least one screen;

second determining whether one of the operably different screen display presentations is displayed on the at least one screen;

when the one operably different screen display presentation is determined to be displayed on the at least one screen, performing a step of second interpreting the second user gesture input for determining a predetermined second operation to apply to the one operably different screen display presentation; and

when none of the operably different screen display presentations are determined in the step of second determining, perform a step of second preventing the device from using the second user gesture input to change the at least one display screen.

15. The computer readable medium of Claim 14, wherein the step of interpreting is for rearranging of a stack for the screen display presentations so that an order of presentation of the screen display presentations is rearranged.

16. The computer readable medium of Claim 14, wherein each of the screen display presentations includes an application window.

17. The computer readable medium of Claim 14, wherein the step of second interpreting is for rearranging of a stack for the operably different screen display presentations so that an order of presentation of the operably different screen display presentations is rearranged.

18. The computer readable medium of Claim 17, wherein each of the operably different screen display presentations includes a desktop.

19. A hand-held device, comprising:

at least one display screen for displaying one or more of screen display presentations and for displaying one or more operably different screen display presentations;

a gesture capture area associated with the at least one screen for receiving first user gesture input, the gesture capture area being separate from the at least one display screen;

wherein the screen display presentations are operably manipulated via the first user gesture input received by the gesture capture area;

wherein the at least one display screen is responsive to user touch for receiving second user gestures input for manipulating the operably different screen presentations;

a display manager for determining: (a) whether one of the screen display presentations is displayed on the at least one screen, and (b) whether one of the operably different screen display presentations is displayed on the at least one screen;

a gesture interpreter for interpreting each of the first and second user gesture inputs for determining an action to perform on a corresponding displayed one of the screen display presentations or a corresponding displayed one of the operably different screen display presentations, wherein the gesture interpreter compares each of the first and second user gesture inputs with a predetermined collection of a plurality of user gesture input descriptions for recognizing user gesture inputs that are interpretable;

wherein for the first user gesture input, if the one screen display presentation is determined to be displayed on the at least one screen, the interpreter interprets the first user gesture input for determining a predetermined first task to apply to the one screen display presentation;

wherein when none of the one or more screen display presentations are determined to be displayed in the display manger prevents the device from using the first user gesture input to change the at least one display screen;

wherein for the second user gestures input, if the one operably different screen display presentation is determined to be displayed on the at least one screen, the interpreter interprets the second user gesture input for determining a predetermined second task to apply to the one operably different screen display presentation; and

wherein when none of the operably different screen display presentations are determined to be displayed in the display manger prevents the device from using the second user gesture input to change the at least one display screen.

a task manager for performing the predetermined first and second tasks.

20. The device of Claim 19, wherein the gesture capture area is not capable of displaying any of the screen display presentations, and is not capable of displaying any of the operably different screen display presentations.

21. The device of Claim 19, wherein the one screen display presentation includes an application window.

22. The method of Claim 19, wherein the one operably different screen display presentation includes a desktop.

ABSTRACT

An intuitive technique for inputting user gestures into a handheld computing device is disclosed allowing a user to better manipulate different types of screen display presentations, such as desktops and application windows, when performing tasks thereon, e.g., minimization, maximization, moving between display screens, and increasing/decreasing a display thereof across multiple display screens. For manipulating an application window on a device display screen for performing tasks as identified above, user gestures are input to a corresponding gesture capture area for this display screen, wherein this capture area is separate from this display screen.

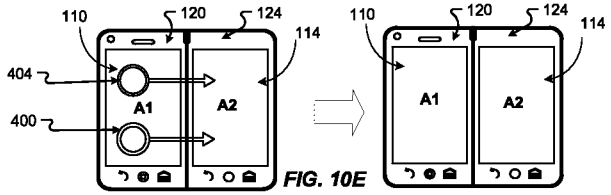


FIG. 10E

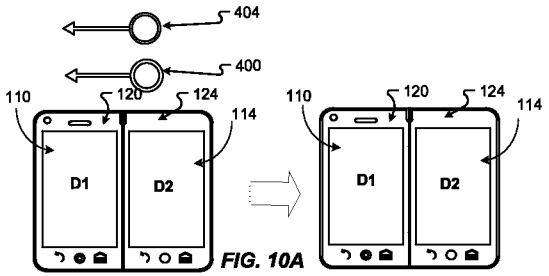


FIG. 10A

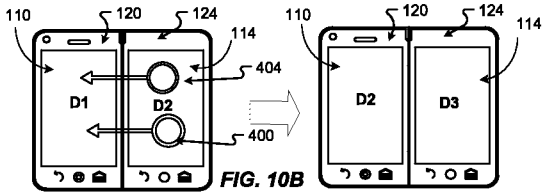


FIG. 10B

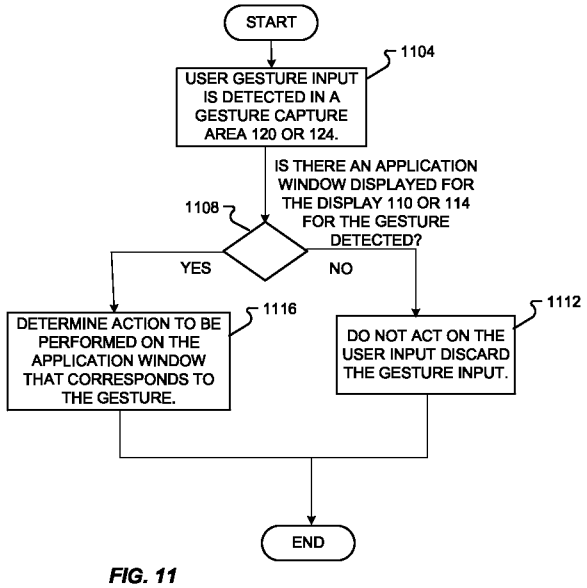


FIG. 11

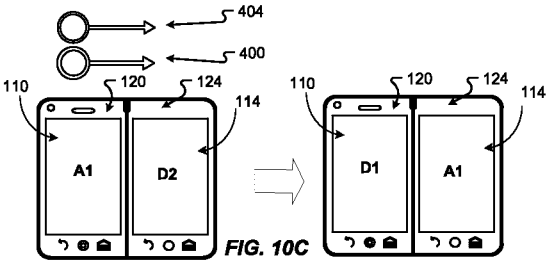


FIG. 10C

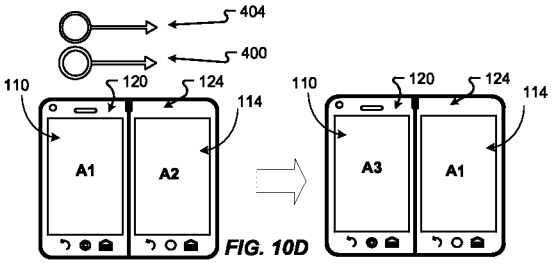


FIG. 10D

METHOD AND SYSTEM FOR VIEWING STACKED SCREEN DISPLAYS USING GESTURES

SUMMARY

There is a need for a dual multi-display handheld computing device that provides for enhanced power and/or versatility compared to conventional single display handheld computing devices. These and other needs are addressed by the various aspects, embodiments, and/or configurations of the present disclosure. Also, while the disclosure is presented in terms of exemplary embodiments, it should be appreciated that individual aspects of the disclosure can be separately claimed.

In at least one embodiment, the present multi-display device includes the following components:

(a) at least one display screen for displaying a plurality of display objects, each object being one of a window or desktop;

(b) a gesture capture area associated with the at least one display screen for receiving first user finger gesture input, the gesture capture area being separate from the at least one display screen;

wherein the at least one display screen is responsive to user touch for receiving a second user finger gesture input;

(c) a display manager for determining which of the plurality of display objects is displayed on the at least one screen;

(d) a gesture interpreter for interpreting each of the first and second user finger gesture inputs for determining an action to perform on the at least one screen display, wherein the gesture interpreter compares each of the first and second user finger gesture inputs with a predetermined collection of a plurality of user gesture input descriptions for recognizing user gesture inputs that are interpretable;

(e) a non-transient data storage for storing data indicative of an ordering of the plurality of display objects;

wherein for a currently displayed one of the display objects when the first user gesture input is received, determining a window or desktop to present on the at least one display screen, wherein the window or desktop is identified as immediately preceding or immediately following, in the ordering the one display object;

wherein the display manager replaces the one display object on the at least one display screen with a display of the window or desktop; and

wherein the ordering is unchanged after the replacement of the one display object with the display of the window or desktop.

Further, in at least one embodiment, the present multi-display device performs the following steps for configuring at least one display screen for displaying a plurality of display objects, each object being one of a window or desktop:

(1) receiving a first user finger gesture input to a gesture capture area operatively associated with the at least one screen, but being separate therefrom;

(2) interpreting the user finger gesture input as a request for navigating an ordering of the plurality of display objects, data for the ordering stored in a non-transient data storage;

(3) in response to the step of interpreting, determining a window or desktop to present on the at least one display screen, wherein the window or desktop is identified as immediately preceding or immediately following, in the ordering, a currently displayed one of the display objects;

(4) as result of the step of determining, performing the steps of (a-1) and (a-2):

(a-1) ceasing to display the currently displayed display object on the at least one screen display; and

(a-2) displaying the window or desktop on the at least one screen display;

wherein after the step of displaying, the ordering is unchanged from before the step of determining.

The present disclosure can provide a number of advantages depending on the particular aspect, embodiment, and/or configuration. In particular, an intuitive technique for inputting user gestures into a handheld computing device is provided that allows a user to better manipulate different types of screen display presentations, such as desktops and application windows, when performing tasks thereon, wherein a window stack for application windows and/or desktops can be navigated and sequentially displayed according to the window stack ordering without disturbing or changing this ordering.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 10 shows a simple series of user finger gestures for navigating through a window stack;

FIG. 11 is a flowchart of an embodiment of high level steps performed when a user input particular finger gestures for navigating through a window stack.

DETAILED DESCRIPTION

In at least one embodiment of the device 100, the MDM 524 (and its Gesture Module 576) may be operable for interpreting gestures in the gesture capture areas 120 and 124 so that such gestures may be used to manipulate the window stack (e.g., 728 described hereinabove). In particular, such manipulations can allow a user to cycle through such a window stack without disturbing or changing the window stack ordering. For example, referring to Fig. 10, window/desktops 1, 2, and 3 are shown as ordered in a window stack, wherein window/desktop 1 is uppermost in the window stack, followed by window/desktop 2, followed by window/desktop 3. As shown, in the first row of Fig. 10, window/desktop 1 consumes the entire display 110 while window/desktop 2 and window/desktop 3 are hidden (as reflected in Fig. 10 the dashing of these windows/desktops). However, this first row also indicates, a user inputs a finger gesture such as the flick gesture 404 or the drag gesture 400 to the gesture capture area 120, wherein such input may be interpreted by the Gesture Module 576 as a request to move the window/desktop 1 off the display 110 and move the window/desktop 2 onto the display 110, but without disturbing or changing the window stack. Thus, a resulting configuration is shown in the second or middle row of Fig. 10, wherein window/desktop 2 is displayed on the display 110. Additionally, this second row of Fig. 10 also shows the user inputting another finger gesture into the gesture capture area 120 for moving the window/desktop 2 off the display 110 and moving the window/desktop 3 onto the display, again without disturbing or changing the order of the windows and/or desktops of the window stack. Accordingly, the third or lowest row of Fig. 10 results, wherein window/desktop 3 is displayed.

Assuming that the windows/desktops 1, 2 and 3 constitute all windows/desktops in the window stack (at least for display 110), the user finger gesture input to the gesture capture area 120 as shown in the third row of Fig. 10, in one embodiment, results in the configuration of the first row of Fig. 10, again without disturbing or changing the ordering in the window stack.

In embodiments, when the user navigates through the window stack (e.g., for the display 110) and thereby displays a window/desktop that is not the uppermost window/desktop of the window stack as in the window stack representation 728 of Fig. 7B, the user can force such a displayed (lower ordered) window/desktop to become the uppermost window/desktop in the window stack by, e.g., providing input directly to this displayed window/desktop, or providing some other input (such as a one or more taps to the display this lower ordered window/desktop). Accordingly, in one embodiment, such a displayed lower ordered window/desktop receiving a user gesture input (or other input), is repositioned to the top of the window stack without disturbing or changing the relative ordering of the other windows/desktops in the window stack. However, alternative rearrangement techniques of the windows/desktops of the window stack are within the scope of the present disclosure when the user provides input to causing a lower ordered window/stack to become the highest. For example, such an alternative rearrangement technique may include reordering the windows/desktops that are higher in the window stack (i.e., higher than the displayed lower ordered window/desktop receiving input) to the lowest orderings in the window stack, but keeping such windows/desktops in their same relative stack order. For example, referring to Fig. 10 again, if such an alternative rearrangement technique were operable, then if the user supplied an input, e.g., directly to the displayed window/desktop 3 (thereby causing this window/desktop to become uppermost in the window stack, then window/desktop 1 would be ordered immediately below window/desktop 3, and window/desktop 2 would be the lowest ordered window/desktop of the window stack.

It should be noted that although the user gestures shown in Fig. 10 show the user navigating down the window stack ordering, a user can also navigate up the window stacking ordering as well. In one embodiment, the same user finger gesture for navigating down the window stack ordering may be used for navigating up the window stack ordering, except that for navigating up the window stack the user gesture(s) therefor may move in an opposite direction. Thus, referring to the middle row of Fig. 10, if the user input the user gesture 400, or 404 in the opposite direction, then window/desktop 1 would be displayed as a result.

Although the above description related to Fig. 10 for navigating a window stack is illustrated for the single display 110, such a navigation technique also applies to display 114. Moreover, a similar navigation technique can be applied when the device 100 is in an open dual-screen state such as 348. In one embodiment, when the user has the device 100 in such an open dual-screen state, a user finger gesture for such navigation may be applied to either of the gesture capture areas 120 or 124, wherein the application window(s) and/or desktops being presented may move across the displays 110, 114 according to, e.g., the following rules:

(a) If a dual-screen application is displayed on both screens, and a user finger gesture is input to one of the gesture capture areas 120, 124 for navigating the window stack (as in Fig. 10), if the next window/desktop (W_1) in the window stack order is to be displayed on both displays 110, 114, then display this next window/desktop W_1 on both displays. Alternatively, if the window/desktop W_1 is to be displayed on a single display, then move it onto one of the displays and also move the next window/desktop (W_2) after W_1 onto one of the displays 110, 114 (converting W_2 to a single display if necessary).

(b) If two single screen windows/desktops (W_1, W_2) are displayed on the displays 110, 114 with W_1 being lower (higher) than W_2 in the window stack ordering, and a user finger gesture is input to one of the gesture capture areas 120, 124 for navigating down (up) the window stack, then one of the windows/desktops W_1, W_2 will move in the direction of the user gesture to replace the other of the window/desktop W_1, W_2, and the (any) next lower (upper) window/desktop (W_3) represented in the window stack will be displayed on the display 110, 114 from which the one window/desktop moved from. Moreover, if W_3 is a dual-screen application/desktop, it will be converted to a single screen display for presentation during such window stack navigation.

Fig. 11 shows a high level flowchart representative of the method for navigating the window stack described hereinabove. In particular, in step 1104, a user gesture is detected in one of the gesture capture areas 120, 124, wherein the gesture is determined by the Gesture Module 576 to be a user request to navigate the window stack and display the windows/desktops according to their ordering in the window stack. Assuming the Gesture Module 576 also determines from the input gesture, the direction the user requests to navigate the window stack

(i.e., up the window stack ordering, or down the window stack ordering), in step 1108, a determination is made by the MDM 524 as to whether there is a next window/desktop represented in the window stack (e.g., 728) for the user indicated direction in the window stack ordering. If the present step determines that there is a next window/desktop (W) represented according to the window stack ordering (e.g., either the next higher window/desktop in the ordering if the user is navigating up the window stack, the next lower window/desktop in the ordering if the user is navigating down the window stack), then step 1112 is performed wherein each of the one or more windows/desktops displayed on the displays 110, 114 (e.g., only a single window/desktop displayed on the active one of the displays 110, 114 when the device 100 is closed, and up to two window/desktops displayed on the displays 110, 114 when the device 100 is in an open state) is shifted so that one of the currently displayed window/desktops moves off the displays 110, 114 entirely, and the next window/desktop W is now displayed on at least one of the displays 110, 114. More precisely, when the device 100 is in the closed state, the operation of moving a currently displayed window/desktop off, and the operation of displaying the window/desktop W occur on the same display 110, 114; alternatively, when the device 100 is in the open state, the operation of moving a currently displayed window/desktop off, and the operation of displaying the window/desktop W occur on different ones of the displays 110, 114. Moreover, if both displays 110, 114 are active (e.g., the device 100 in an open state), assuming there is a different window/desktop being displayed on each of the displays 110, 114, then as a first of the windows/desktops shifts entirely off the displays 110, 114, a second of the displayed window/desktop shifts to the display to the display that had the first window/desktop, and the next window/desktop W is displayed on the display 110, 114 that was displaying the second window/desktop. Alternatively, if it is determined in step 1108 that there is no next window/desktop (i.e., a currently window/desktop being displayed is the top of the window stack, or the bottom of the window stack), then step 1116 is performed wherein, in the present embodiment, the window/desktop for the opposite end of the window stack is displayed by shifting a currently displayed window/desktop off the display(s) 110, or 114 in a manner similar to that of step 112. In particular, if the device 100 is in an open state with two window/desktops displayed (one per display), then the window/desktop currently displayed but not shifted off the displays 110, 114 is shifted to the other display.

Regardless of which of the steps 1112 or 1116 is performed, in step 1120 a determination is made by the MDM 524 as to whether another user gesture is detected in the gesture capture area 120 or 124 for continuing to navigate through the windows/desktops for the window stack. Accordingly, if such an additional user gesture is detected, then step 1108 is again performed. Alternatively, if no additional window stack navigation gesture is detected in the gesture capture area 120, 124 (e.g., the device 100 is instead opened, or closed, or a different user input is provided to one of the gesture capture areas, or a user input is provided to one of the displays 110, 114), then the navigation of the window stack ceases. Note that since the window stack ordering of its windows/desktops did not change, unless the user supplied an input to one of the windows/desktops being displayed via the window stack navigation process of Fig. 11, if the device 100 is, e.g., turned off and then turned back on, the window(s)/desktop(s) shown on the active display(s) 110 or 114 will be the window(s)/desktop(s) at the top of the window stack.

What is claimed is:

1. A method for configuring a device having at least one display screen for displaying a plurality of display objects, each object being one of a window or desktop, comprising:

receiving a first user finger gesture input to a gesture capture area operatively associated with the at least one screen, but being separate therefrom;

interpreting the user finger gesture input as a request for navigating an ordering of the plurality of display objects, data for the ordering stored in a non-transient data storage;

in response to the step of interpreting, determining a window or desktop to present on the at least one display screen, wherein the window or desktop is identified as immediately preceding or immediately following, in the ordering, a currently displayed one of the display objects;

as result of the step of determining, performing the steps of:

(a-1) ceasing to display the currently displayed display object on the at least one screen display; and

(a-2) displaying the window or desktop on the at least one screen display.

2. The method of Claim 1, wherein the gesture capture area includes a capacitive touch sensitive area for receiving the user gesture input.

3. The method of Claim 2, wherein the gesture capture area is not capable of displaying any of the display objects.

4. The method of Claim 1, wherein after the step of displaying, the ordering is unchanged from before the step of determining.

5. The method of Claim 4, comprising:

second receiving a second user finger gesture input to the gesture capture area associated with the at least one screen;

second interpreting the second user finger gesture input as a second request for navigating the ordering of the plurality of display objects;

second determining a second window or desktop to present on the at least one display, wherein the second window or desktop is identified as immediately preceding or immediately following, in the ordering, the window or desktop;

as result of the step of second determining, performing the steps of:

(b-1) second ceasing to display the window or desktop on the at least one screen display; and

(b-2) second displaying the second window or desktop on the at least one screen display.

6. The method of Claim 5, wherein the at least one display screen includes a plurality of display screens, and the window or desktop displays on a second of the display screens prior to a completion of the step of second displaying.

7. The method of Claim 5, wherein each of the user gesture input and the second user gesture input is interpreted as being for a same user gesture.

8. The method of Claim 7, wherein the user gesture input includes one of: a drag, and a flick.

9. The method of Claim 4, further including:

receiving a user finger gesture input to the displayed the window or desktop on the at least one screen display;

changing the ordering as a result of the user finger gesture input;

second receiving a second user finger gesture input to the gesture capture area associated with the at least one screen;

second interpreting the second user finger gesture input as a second request for navigating the changed ordering of the plurality of display objects;

second determining a second window or desktop to present on the at least one display, wherein the second window or desktop is identified as immediately preceding or immediately following the window or desktop in the changed ordering;

as result of the step of second determining, performing the steps of:

(b-1) second ceasing to display the window or desktop on the at least one screen display; and

(b-2) second displaying the second window or desktop on the at least one screen display.

10. The method of Claim 1, wherein there is a predetermined collection of a plurality of user finger gesture input descriptions for recognizing user gesture inputs that are interpretable in the step of interpreting.

1. 11. A computer readable medium for configuring a device having at least one display screen for displaying a plurality of display objects, each object being one of a window or desktop, comprising:

2. machine instructions for performing the following steps:

receiving a first user finger gesture input to a gesture capture area operatively associated with the at least one screen, but being separate therefrom;

interpreting the user finger gesture input as a request for navigating an ordering of the plurality of display objects, data for the ordering stored in a non-transient data storage;

in response to the step of interpreting, determining a window or desktop to present on the at least one display screen, wherein the window or desktop is identified as immediately preceding or immediately following, in the ordering, a currently displayed one of the display objects;

as result of the step of determining, performing the steps of:

(a-1) ceasing to display the currently displayed display object on the at least one screen display; and

(a-2) displaying the window or desktop on the at least one screen display.

13. The computer readable medium of Claim 12, wherein the step of interpreting is for navigating a window stack for the plurality of display objects so that the ordering of the display objects remains unchanged.

3. 14. The computer readable medium of Claim 13, wherein the machine instructions further perform the following steps:

second receiving a second user finger gesture input to the gesture capture area associated with the at least one screen;

second interpreting the second user finger gesture input as a second request for navigating the ordering of the plurality of display objects;

second determining a second window or desktop to present on the at least one display, wherein when the window or desktop is identified as a top or bottom display object in the ordering, the second window or desktop is identified as, respectively, being a bottom display object in the ordering, or a top display object in the ordering;

as result of the step of second determining, performing the steps of:

(b-1) second ceasing to display the window or desktop on the at least one screen display; and

(b-2) second displaying the second window or desktop on the at least one screen display.

15. The computer readable medium of Claim 14, wherein the machine instructions further perform the following steps:

receiving a user finger gesture input to the displayed the second window or desktop on the at least one screen display; and

changing the ordering as a result of the user finger gesture input.

16. A hand-held device, comprising:

at least one display screen for displaying a plurality of display objects, each object being one of a window or desktop;

a gesture capture area associated with the at least one display screen for receiving first user finger gesture input, the gesture capture area being separate from the at least one display screen;

wherein the at least one display screen is responsive to user touch for receiving a second user finger gesture input;

a display manager for determining which of the plurality of display objects is displayed on the at least one screen;

a gesture interpreter for interpreting each of the first and second user finger gesture inputs for determining an action to perform on the at least one screen display, wherein the gesture interpreter compares each of the first and second user finger gesture inputs with a predetermined collection of a plurality of user gesture input descriptions for recognizing user gesture inputs that are interpretable;

a non-transient data storage for storing data indicative of an ordering of the plurality of display objects;

wherein for a currently displayed one of the display objects when the first user gesture input is received, determining a window or desktop to present on the at least one display screen, wherein the window or desktop is identified as immediately preceding or immediately following, in the ordering the one display object;

wherein the display manager replaces the one display object on the at least one display screen with a display of the window or desktop; and

wherein the ordering is unchanged afrer the replacement of the one display object with the display of the window or desktop.

17. The device of Claim 16, wherein the gesture capture area is not capable of displaying any of the display objects.

18. The device of Claim 16, further including a task manager, wherein when data indicative of the second user finger gesture is input is received by the display manager afrer the window or desktop is displayed, task manager changes the ordering so that the window or desktop is represented as being at a top of the ordering.

19. The device of Claim 16, wherein the at least one display screen includes a plurality of display screens, and when the display manager replaces the one display object on a first of the display screens with a display of the window or desktop, the display manager displays the one display object on a second of the display screens so that the one display object and the window or desktop display concurrently.

20. The device of Claim 19, further including:

an attachment configured to foldably connect the first display screen with the second display screen;

a first sensor configured to produce a first signal indicative of a fold position of the first display screens relative to one another; and

a computer-readable medium having instructions stored thereon that include:

a first set of the instructions configured to determine that the device has moved from a first physical orientation to a different second physical orientation, wherein the first and second physical orientations differ by the first physical orientation being one of: folded or in a landscape orientation;

wherein the ordering does not change as a result of folding the device, and the ordering does not change as a result of unfolding the device.

ABSTRACT

An intuitive technique for inputting user gestures into a handheld computing device is disclosed allowing a user to better manipulate different types of screen display presentations, such as desktops and application windows, when performing tasks thereon, wherein a window stack for application windows and/or desktops can be navigated and sequentially displayed according to the window stack ordering without disturbing or changing this ordering.

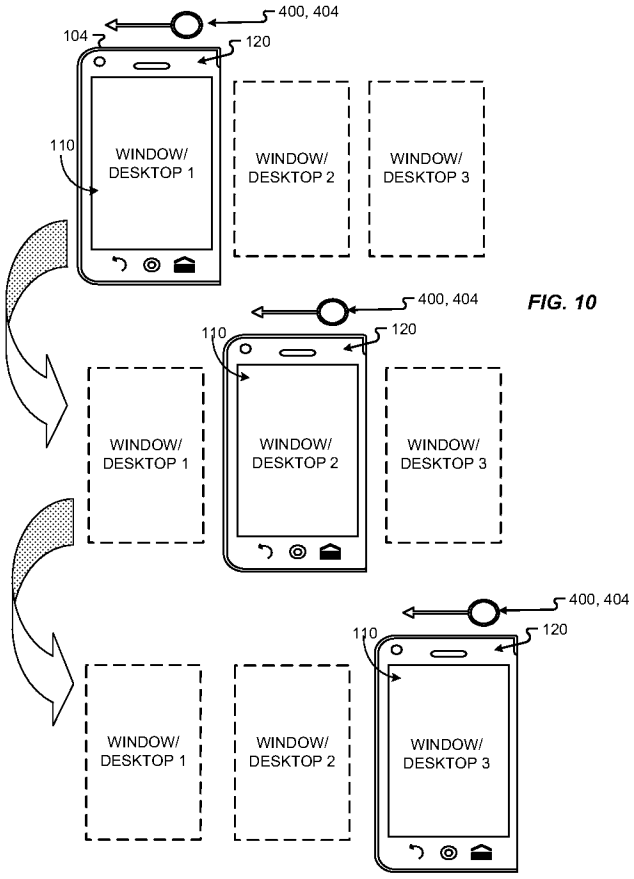

400, 404
104
120
110
WINDOW/ DESKTOP 1
WINDOW/ DESKTOP 2
WINDOW/ DESKTOP 3
400, 404
110
120
WINDOW/ DESKTOP 1
WINDOW/ DESKTOP 2
WINDOW/ DESKTOP 3
400, 404
110
120
WINDOW/ DESKTOP 1
WINDOW/ DESKTOP 2
WINDOW/ DESKTOP 3

FIG. 10

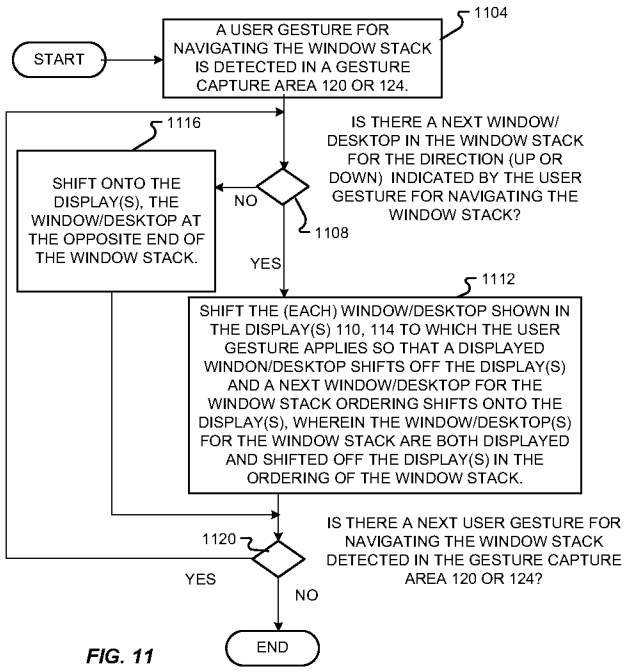

START
A USER GESTURE FOR NAVIGATING THE WINDOW STACK IS DETECTED IN A GESTURE CAPTURE AREA 120 OR 124.
1104
IS THERE A NEXT WINDOW/ DESKTOP IN THE WINDOW STACK FOR THE DIRECTION (UP OR DOWN) INDICATED BY THE USER GESTURE FOR NAVIGATING THE WINDOW STACK?
1108
NO
1116
SHIFT ONTO THE DISPLAY(S), THE WINDOW/DESKTOP AT THE OPPOSITE END OF THE WINDOW STACK.
YES
1112
SHIFT THE (EACH) WINDOW/DESKTOP SHOWN IN THE DISPLAY(S) 110, 114 TO WHICH THE USER GESTURE APPLIES SO THAT A DISPLAYED WINDON/DESKTOP SHIFTS OFF THE DISPLAY(S) AND A NEXT WINDOW/DESKTOP FOR THE WINDOW STACK ORDERING SHIFTS ONTO THE DISPLAY(S), WHEREIN THE WINDOW/DESKTOP(S) FOR THE WINDOW STACK ARE BOTH DISPLAYED AND SHIFTED OFF THE DISPLAY(S) IN THE ORDERING OF THE WINDOW STACK.
1120
IS THERE A NEXT USER GESTURE FOR NAVIGATING THE WINDOW STACK DETECTED IN THE GESTURE CAPTURE AREA 120 OR 124?
YES
NO
END

FIG. 11

BRIEF DESCRIPTION OF DRAWINGS

FIG. 10A depicts a first display state of an open device with various non-displayed windows stacked on one side of the visible display in accordance with embodiments of the present disclosure;

FIG. 10B depicts a second display state of an open device with various non-displayed windows stacked on either side of the visible display in accordance with embodiments of the present disclosure;

FIG. 10C depicts a third display state of an open device with various non-displayed windows stacked on the other side of the visible display as compared to the display of Fig. 10A in accordance with embodiments of the present disclosure.

FIGS. 10A-C depict graphical representations of embodiments of a device 100 in an open portrait state where the primary screen or display 104 in a first portrait configuration 604 is shown surrounded by a solid line, while the secondary screen or display 108 is in a second portrait configuration 608 and is also visible as indicated by a solid line.

DETAILED DESCRIPTION

Referring first to Fig. 10A, the primary screen 104 shows a first page 1004 showing a first desktop D1, and the secondary display 108 shows a first page 1006 of a single page application. Two other pages 1012 and 1014 that are not visible on the primary screen 104 or the secondary screen 108 are also graphically represented as surrounded by dashed lines. Figure 10A further shows the present device 100 where a desktop reveal expansion has been initiated. The desktop reveal expansion may be initiated by, (1) a user input gesture, (2) combination of user input gestures, (3) a memory output, (4) a response to a predetermined condition (e.g., application control, power levels, communications interrupt, operating system state, device screen state open/closed, timers, and single or multiple sensor outputs) or (5) any combination thereof, such that the MDM class 524 registers and interprets as an initiation command. As noted above, the graphical representations of the desktops and running applications in the pages can conceptually be thought of as one or more combinations of virtual "stacks." These stacks of pages can be created, deleted, shuffled, and/or moved by a user or by an output from the MDM class 524, similar to a user manipulating the virtual playing cards of a deck stack.

The concept of stacks disclosed herein may also be applied to desktops available on the device 100. In other words, a desktop virtual stack may be used for the desktops which can be displayed by the device 100 and an application virtual stack may be used for the applications which can be displayed by the device 100. Specifically, when a desktop is revealed it may be divided into multiple pages for display on multiple screens of the device 100. These pages may be manipulated in their own desktop virtual stack, or stacks, separate from the application virtual stacks and/or other virtual stacks. This separation of stacks allows for greater user flexibility in navigating through different applications or desktops by creating an intuitive interface dependent on the data displayed to the screens 104, 108. However, it is also anticipated that these stacks could be combined to form a single virtual stack. Further, pages in each stack may move together as if the pages in a stack are virtually connected. Therefore, a single position move of a page in one stack correspondingly moves all of the pages in that stack by one position.

In FIGS. 10A-C, the stacks of individual pages 1004, 1008, 1012, and 1016 are shown spread outwardly in a linear format on both the left side of the primary screen 104 as well the right side of the non-visible secondary screen 108 to better visualize the organization of the virtual stack. The position of a page from the primary screen 104 or secondary screen 108 is the position of the page in each respective stack stored on the device 100. Therefore, if a page is graphically represented immediately adjacent to the primary screen 104, or immediately adjacent to the secondary screen 108, it is the first page in the stack that may be moved onto that adjacent screen with a user input gesture. Although only several pages of a stack are shown in FIGS. 10A-C, the number of pages available to be organized in the stack can vary and, in some embodiments, may equal the number of applications, desktops, and/or other displayable information created on the device 100. Moreover, the stack adjacent to the primary screen 104 and the stack adjacent to the secondary screen 108 may act as a unified stack (where a move or manipulation of a page from one stack moves or manipulates the pages in both stacks), or as independent stacks (each screen having its own independently navigable stack of pages).

FIG. 10B shows an embodiment of the present device 100, in an open portrait state, where the desktop reveal expansion has been initiated and the stack has been indexed to display the first page application of page 1008 on the primary screen 104, and the second page application of page 1012 on the secondary screen 108. Accordingly, page 1004 previously showing the first desktop D1 is now non-visible, and the previously non-visible page 1012 is now visible on the secondary screen 108. Conceptually, this manipulation of the stack is shown in this figure as a unitary movement of all of the pages in the stack to the left by one position increments. In other words, the manipulation of the desktops/applications by a gesture allows the user to move the desktops/applications to either a new position on the other screen, or a new position in which the selected desktop/application is moved to a non-visible position. While the desktop reveal expansion is initiated, the user can navigate between desktop and application pages in a stack by moving them into view on the primary screen 104 and/or secondary screen 108 by providing a user input gesture 1020. This input may be detected by one or more of the hardware components of the device 100 as discussed herein. The desktops may move together when being navigated by the user. This linked movement of desktops is similar to embodiments in which the pages in a stack can move together as disclosed herein. Examples of input gestures 1020 that can be used include an off-screen drag gesture or an off-screen flick gesture.

FIG. 10C shows an embodiment of the present device 100, in an open portrait state, where the desktop reveal expansion has been initiated and the stack has been further indexed by a one position increment to display the second page application of page 1012 on the primary screen 104, and the second desktop D2 of page 1014 on the secondary screen 108. Accordingly, page 1004 is still non-visible, the previously visible page 1008 is now non-visible, and the second desktop D2 is now visible on the secondary screen 108.

Although Figs. 10A-10C are presented in the open portrait state, it shall be understood that the features disclosed in these figures can also be incorporated in other modes of operation of the device 100 to include a landscape state or any of the other "physical" states disclosed herein. Based on the description of Figs. 10A-10C, it is evident that the display of windows in a window stack can be selectively changed by simple gesture interactions thereby making multiple desktops and running applications available to the user with minimum effort. Each screen has a logical representation of the windows that have been opened and displayed on the screens. The logical representation is the stack that is made up of one or more windows arranged in an order that can be selected and changed by the user. The windows can be indexed or shifted to display desire windows, as well as to move windows between separate stacks. The ordered arrangement of a stack maintains the logical construction of the stack intact when the user navigates the window stack, for example in an off-screen area. Thus, the device operates in a very intuitive way for the user to simultaneously manage multiple desktops and applications.

CLAIMS

1. A method of displaying a plurality of desktops and/or applications, each having at least one window, on a multi-screen device, comprising:

receiving, by a processor, a first input that represents an instruction to reveal one of a desktop or application on a first display of the multi-screen device;

selecting, by a processor, a first desktop or application to display on the first display;

receiving, by a processor, a second input that represents an instruction to reveal one of a desktop or application on a second display of the multi-screen device;

selecting, by a processor, a second desktop or application to display on the second display;

displaying, by a processor, the selected first and second desktops or applications on the first and second displays; and

selectively moving , by a processor, at least one of the displayed first and second desktops or applications to another selected position comprising either a position on the other display or a non-visible position.

2. The method of claim 1, wherein the first and second inputs comprise at least one of the following: i) a user input gesture; ii) a combination of user input gestures; iii) a memory output; iv) a response to a programmed condition; and v) hardware timers.
3. The method of claim 1, wherein the selectively moving step is achieved by at least a user input gesture.
4. The method of claim 3, wherein the user input gesture comprises at least one of a drag gesture or a flick gesture.
5. The method of claim 3, wherein the drag gesture and flick gesture are conducted on off-screen areas of the first and second displays.
6. The method of claim 1, wherein the selected first and second desktops or applications on the first and second displays are arranged in at least one window stack.
7. The method of claim 6, wherein the user manipulates the order of the window stack by at least a user input gesture that corresponds to a logical order of the window stack in which the user selectively moves the first and second desktops or applications to either visible positions on the first and second displays or to a non-visible position.

8. A non-transitory computer-readable medium having stored thereon instructions that cause a computing system to execute a method, the instructions comprising:

instructions configured to receive a first predetermined input that represents an instruction to reveal a desktop or application on a first display of the multi-screen device;

instructions configured to determine a first desktop or application to display on the first display;

instructions configured to receive a second predetermined input that represents an instruction to reveal a desktop or application on a second display of the multi-screen device;

instructions configured to determine a second desktop or application to display on a second display of the multi-screen device; and

instructions configured to respond to the first and second predetermined inputs with outputs that cause the first desktop or application to be displayed on the first display and cause the second desktop or application to be displayed on the second display

instructions configured to receive a third predetermined input that represents an instruction to change the visible position of the first and second desktops or applications;

instructions configured to respond to the third predetermined input that causes the first and second desktop and application to change locations to either the other display or to a non-visible position.

9. The computer-readable medium of claim 8, wherein the first display corresponds to a primary screen and wherein the second display corresponds to a secondary screen.

10. The computer-readable medium of claim 8, further comprising: instructions configured to receive a fourth predetermined input that represents an instruction to change the position of a third non-visible desktop or application; and instructions configured to respond to the fourth predetermined input that causes the third non-visible desktop or application to move to a visible position on either the first or second displays thereby replacing the previously displayed desktop or application.

11. The computer-readable medium of claim 8, further comprising:

instructions configured to receive a fourth predetermined input that represents an instruction to change the position of one of the first or second desktops or applications from a visible position on one of the first or second displays to the non-visible position;

and instructions configured to respond to the fourth predetermined input that causes the first or second desktops or applications to move to the non-visible position.

12. The computer-readable medium of claim 8, wherein:

the third predetermined input comprises at least a user input gesture.

13. The computer-readable medium of claim 8, wherein the first and second desktops or applications on the first and second displays are arranged in at least one window stack.

14. The computer-readable medium of claim 13, wherein the user manipulates the order of the window stack by at least a user input gesture that corresponds to a logical order of the window stack in which the user selectively moves the first and second desktops or applications to either visible positions on the first and second displays or to a non-visible position.

15. A multi-screen user device, comprising:

a first display including a first display area;

a second display including a second display area;

a first user input gesture area of the first display;

a second user input gesture area of the second display, wherein the first and second user input gesture areas are configured to accept input from a user.

a computer-readable medium having instructions stored thereon that include:

a first set of instructions configured to determine a number and identity of desktops or applications selected to be run by a user;

a second set of instructions configured to determine, based on the number and identity of the desktops or applications running, a window stack comprising a logical representation of the desktops and applications within an ordered group whereby a user can selectively move a selected desktop or application to either a visible position on another display or a non-visible position; and

a third set of instructions responsive to gestures made by the user to move a perceived position of the selected desktop or application to the selected position as determined by a type of gesture executed by the user.

16. The device of claim 16, wherein the first display corresponds to a primary screen and wherein the second display corresponds to a secondary screen.

17. The device of claim 16, wherein the first user input gesture area is located on an off-screen area of the first display and outside of the first display area.

18. The device of claim 16, wherein the second user input gesture area is located on an off-screen area of the second display and outside of the second display area.

19. The device of claim 16, wherein the first and second displays show different displayed images by user input gestures that manipulate an order of the desktops and applications in the window stack.

20. The device of claim 19, wherein the user input gestures executed correspond to a logical order of the window stack in which the user selectively moves the first and second desktops or applications to either visible positions on the first and second displays or to a non-visible position

ABSTRACT

A dual-screen user device and methods for revealing a combination of desktops on single and multiple screens are disclosed. Specifically, a determined number of desktops and/or running applications can be selectively displayed on dual screen displays conditioned upon inputs received and the state of the device. Desktop displays and applications can be selectively shifted between the screens by user gestures, and/or selected desktop displays and applications can be moved off of the screens by other user gestures and therefore hidden. The hidden desktops and screens however can be re-displayed by yet another gesture. The desktops and applications are arranged in a window stack that represents a logical order of the desktops and applications therefore providing a user with an intuitive ability to manage multiple applications and desktops running simultaneously by selected user gestures. The user gestures prevent the user from having to open and close the running desktops and applications, or to execute other user commands that otherwise result in a laborious effort by the user to manage the multiple desktops and applications.

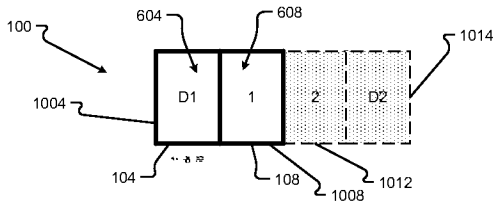


FIG. 10A

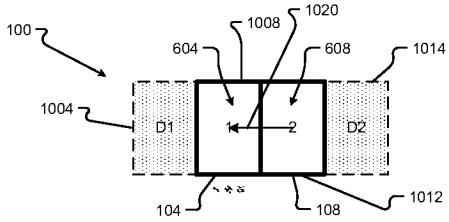


FIG. 10B

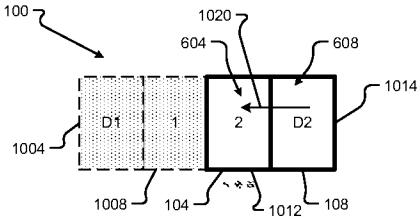


FIG. 10C

Brief Description of Drawings

[0001] FIG. 10A depicts a first display state of an open device with in accordance with embodiments of the present disclosure;

[0002] FIG. 10B depicts a second display state of an open device after execution of a first pinch gesture resulting in the displayed applications swapping or switching screens in accordance with embodiments of the present disclosure; and

[0003] FIG. 10C depicts a third display state of an open device after execution of a second pinch gesture resulting in the displayed applications again swapping or switching screens in accordance with embodiments of the present disclosure;

[0004] Figs. 10A-C depict graphical representations of embodiments of a device 100 in an open portrait state where the primary screen 104 in a first portrait configuration 604 is shown surrounded by a solid line, while the secondary display 108 is in a second portrait configuration 608 and is also visible as indicated by a solid line.

[0005] Detailed Description

[0006] Referring first to Fig. 10A, the primary screen 104 shows a first page 1004 showing a first single page application (1), and the secondary display 108 shows a first page 1008 of another single page application (2). Although FIG. 10A shows exemplary applications (1) and (2), it shall be understood that the pinch gesture can be used to swap or switch any two running applications or desktops displayed as the top applications/desktops on each of the screens. FIG. 10A also shows the symbol 1012 for the pinch gesture indicating that a user has just executed the gesture. The user executes the gesture as more fully described above with respect to FIG. 4

[0007] Referring to FIG. 10B, this figure illustrates that after the first pinch gesture, the applications (1) and (2) have switched or swapped locations to the other screen. Accordingly, instead of the application (1) being on the top of the first screen 104, it is now on top of the second screen 108. Also, instead of the application (2) being on the top of the second screen 108, it is now on top of the first screen 104.

[0008] FIG 10B also shows the symbol 1012 indicating a second pinch gesture is executed. Accordingly now referring to FIG 10C, the applications (1) and (2) have now been swapped or switched back to their original positions as shown in FIG 10A. Thus, it should be understood that a user can selectively and repeatedly swap or switch the locations of applications or desktops between the two screens.

[0009] Although Figs. 10A-10C are presented in the open portrait state, it shall be understood that the features disclosed in these figures can also be incorporated in other modes of operation of the device 100 to include a landscape state or any of the other "physical" states disclosed herein.

[0010] Based on the description of Figs. 10A-10C, it is evident that the display of running desktops or applications can be selectively swapped or switched between screens by a simple gesture interaction thereby enabling the user to easily position and manipulate running desktops/applications in a manner of their choosing.

[0011] Claims

[0012]

[0013] 1. A method of displaying a plurality of desktops and/or applications, each having at least one window, on a multi-screen device, comprising:

receiving, by a processor, a first input that represents an instruction to reveal one of a desktop or application on a first display of the multi-screen device and selecting a first desktop or application to display on the first display;

receiving, by a processor, a second input that represents an instruction to reveal one of a desktop or application on a second display of the multi-screen device and selecting a second desktop or application to display on the second display;

displaying, by a processor, the selected first and second desktops or applications on the first and second displays; and

switching, by a processor, the positions of the displayed first and second desktops or applications by execution of a pinch gesture such that the desktop or application appearing on the first display is switched to the second display, and the desktop or application appearing on the second display is switched to the first display.

[0014] 2. The method of claim 1, wherein the pinch gesture is conducted on off-screen areas of the first and second displays.

[0015] 3. The method of claim 1, wherein the desktops or applications on the first and second displays are arranged in at least one window stack.

[0016] 4. The method of claim 1, wherein execution of the pinch gesture results in a simultaneous switching of the positions of the displayed first and second desktops or applications.

[0017] 5. The method of claim 1, wherein execution of the pinch gesture can be executed a number of times to continue to selectively switch positions of displayed desktops or applications.

[0018] 6. The method of claim 1, wherein the pinch gesture is initiated by a first digit of the user contacting a sensing portion of the first display and a second digit of the user contacting a sensing portion of the second display.

[0019] 7. A non-transitory computer-readable medium having stored thereon instructions that cause a computing system to execute a method, the instructions comprising:

[0020] instructions configured to receive and determine a first predetermined input that represents an instruction to reveal a first desktop or application on a first display of the multi-screen device;

instructions configured to receive and determine a second predetermined input that represents an instruction to reveal a second desktop or application on a second display of the multi-screen device;

instructions configured to respond to the first and second predetermined inputs with outputs that cause the first desktop or application to be displayed on the first display and cause the second desktop or application to be displayed on the second display;

instructions configured to receive a third predetermined input that represents instructions prompted by execution of a pinch gesture to switch the displayed positions of the first and second desktops or applications;

instructions configured to respond to the third predetermined input that causes the first and second desktops and applications to switch the displayed positions, such that the first desktop or application appearing on the first display is switched to the second display, and the second desktop or application appearing on the second display is switched to the first display.

[0021] 8. The computer-readable medium of claim 7, wherein the first display corresponds to a primary screen and wherein the second display corresponds to a secondary screen.

[0022] 9. The computer-readable medium of claim 7, wherein the pinch gesture is conducted on off-screen areas of the first and second displays.

[0023] 10. The computer-readable medium of claim 7, wherein the desktops or applications on the first and second displays are arranged in at least one window stack.

[0024] 11. The computer-readable medium of claim 7, wherein execution of the pinch gesture results in a simultaneous switching of the positions of the displayed first and second desktops or applications.

[0025] 12. The computer-readable medium of claim 7, wherein the pinch gesture can be executed a number of times to continue to selectively switch positions of displayed desktops or applications.

[0026] 13. The computer-readable medium of claim 7 wherein the instructions configured to receive the third input includes instructions configured to respond to the pinch gesture initiated by a first digit of the user contacting a sensing portion of the first display and a second digit of the user contacting a sensing portion of the second display.

[0027] 14. A multi-screen user device, comprising:

a first display including a first display area;

a second display including a second display area;

a first user input gesture area of the first display;

a second user input gesture area of the second display, wherein the first and second user input gesture areas are configured to accept input from a user.

a computer-readable medium having instructions stored thereon that include:

(i) a first set of instructions configured to determine a number and identity of desktops or applications to be displayed on the first and second displays in response to a user input that selects desired desktops or applications to be displayed on the first and second displays;

(ii) a second set of instructions to configured to display the selected desktops or applications such that a first desktop or application is displayed on the first display and a second desktop or application is displayed on the second display

(iii) a third set of instructions responsive to a pinch gesture made by the user to cause the first and second desktops and applications to switch the displayed positions, such that the first desktop or application appearing on the first display is switched to the second display, and the second desktop or application appearing on the second display is switched to the first display.

[0028] 15. The device of claim 14, wherein the first display corresponds to a primary screen and wherein the second display corresponds to a secondary screen.

[0029] 16. The device of claim 14, wherein the first user input gesture area is located on an off-screen area of the first display and outside of the first display area.

[0030] 17. The device of claim 14, wherein the second user input gesture area is located on an off-screen area of the second display and outside of the second display area.

[0031] 18. The device of claim 14, wherein the displayed desktops or applications on the first and second displays are arranged in at least one window stack

[0032] 19. The device of claim 14, wherein execution of the pinch gesture results in a simultaneous switching of the positions of the displayed first and second desktops or applications

[0033] 20, The device of claim 14, wherein execution of the pinch gesture can be executed a number of times to continue to selectively switch positions of displayed desktops or applications

ABSTRACT

A dual-screen user device and methods for revealing a combination of desktops on single and multiple screens are disclosed. Specifically, a determined number of desktops and/or running applications can be selectively displayed on dual screen displays conditioned upon inputs received and the state of the device. Desktop displays and applications can be selectively shifted between the screens by a user gesture in the form of a pinch gesture. This single user gesture prevents the user from having to open and close the running desktops and applications, or to execute multiple other user commands that otherwise would result in a laborious effort by the user to manage the multiple desktops and applications.

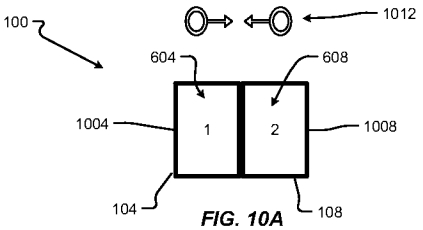


FIG. 10A

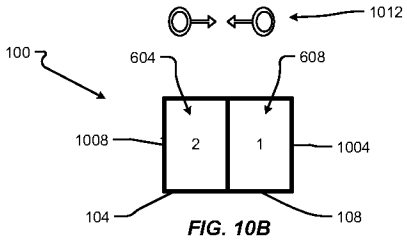


FIG. 10B

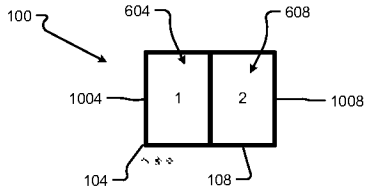


FIG. 10C

[0100] The device 100 manages desktops and/or windows within a user interface 700, as shown in Figs. 7A through 7G. A user interface 700 is a logical arrangement of active and inactive windows for a multi-screen device. For example, the user interface 700 may be logically similar to a deck of cards or a film strip, where one or more windows or desktops are arranged in order, as shown in Fig. 7A. An active window is a window that is currently being displayed on at least one of the touch sensitive displays 110, 114. For example, window 1 704 and window 3 708 are active windows and are displayed on touch sensitive displays 110 and 114, respectively. An inactive window is a window that was opened and displayed but is now "behind" an active window and not being displayed. In embodiments, an inactive window may be for an application that is suspended in a "sleep state", and thus, the window is not displaying active content. For example, window 2 712 is an inactive window in the sleep state.

[0101] A user interface 700 may have various arrangements or organizational structures. In the embodiment shown in Fig. 7A, the device 100 includes a set of windows/display objects associated with a first touch sensitive display 110 and a second set of windows/display objects associated with a second touch sensitive display 114. Thus, each touch sensitive display 110, 114 can have an associated window set of windows/display objects. The two set of windows/display objects may have different numbers of windows and may be arranged differently. Further, the two set of windows/display objects can also be identified differently and managed separately. Thus, the set of windows/display objects can be arranged in order from a first window to a last window 724. Likewise, the second set of windows/display objects can be arranged from a first window to a last window, but may have more or fewer windows. A logical data structure for managing the sets of windows/display objects may be as described in conjunction with Fig. 8.

[0102] When a window is moved, the newly moved window is repositioned within the user interface 700. Where and how the window is positioned within the user interface 700 can be a function of the orientation of the device 100, the context of what programs, functions, software, etc. are being executed on the device 100, how the user interface 700 is arranged when a new window is opened, when a window is moved, etc. To change the window position in the user interface 700, the window is moved by a user and information associated with the window can change. With this information, a logical data structure for the window can be changed and stored. When user interface or other events or tasks change the arrangement of windows, the user interface 700 can be changed to reflect the change in arrangement. It should be noted that these same concepts described above can be used to manage the one or more desktops for the device 100.

[0103] For example, an arrangement of windows in a user interface can be shown in "profile" views in Figs. 7B through 7G. A top view is shown in Figs. 7B and 7G. Side views are shown in Figs. 7C-7F. Here, the logical arrangement of the user interface 700 is shown where, the active windows 704 and 708 are shown in Fig. 7B. The windows 704 and 708 can be the top most windows in portions of a composite display 720, which represents the total area of all the touch sensitive displays 110,114. As shown in Figs. 7C and 7D, window 1 704 and window 3 708 are at the top of the set of windows/display objects. The windows "beneath" the top windows, window 1 704 and window 3 708, are not active. Rather, the window, window 2 712, which is not displayed, is inactive or in a sleep state. In a sleep state, the window 2 712 does not receive inputs. Further, the window need not generate a user interface window. As such, by setting the window 2 712 inactive, the processor saves on processor cycles and required code execution. When a window is "awakened", it is brought back to an active status, begins receiving inputs, and renders a window user interface. As shown in Figs. 7C and 7D, window 2 712 is below window 3 708 and not displayed, which changes in the embodiment shown in Figs. 7E and 7F.

[0104] In Figs. 7E through 7G, window 3 708 is moved from the second touch sensitive display 114 onto a first touch sensitive display 110 by gesture or other input to the device 110. In moving window 3 704, window 1 704 is covered and becomes inactive. Window 2 712 is exposed and becomes active. When a window 1 704 becomes inactive, a bitmap of the last view of the window is stored. The bitmap can be used when windows transfer onto the active display(s). Thus, the window appears to be active when transitioning but is not actually rendering a user interface until the transition is complete, and the window becomes active. Window 2 712 appears as the bitmap until window 3 708 has completed the move to the first touch sensitive display 110, and window 2 becomes active, as shown on Fig. 7G.

[0105] A logical data structure 800 for managing the arrangement of windows or desktops in a window user interface 700 is shown in Fig. 8. The logical data structure 800 can be any data structure used to store data whether an object, record, file, etc. The logical data structure 800 can be stored in any type of database or data storage system, regardless of protocol or standard. In

embodiments, the logical data structure 800 includes one or more portions, fields, attributes, etc. that store data in a logical arrangement that allows for easy storage and retrieval of the information. Hereinafrer, these one or more portions, fields, attributes, etc. shall be described simply as fields. The fields can store data for a window identifier 804, a dimension 808, a window stack position identifier 812, a display identifier 816, and/or an active indicator 820. Each window in a window user interface 700 can have an associated logical data structure 800. While only a single logical data structure 800 is shown in Fig. 8, there may be more or fewer logical data structures 800 used with a window user interface 700 (based on the number of windows or desktops in the user interface 700), as represented by ellipses 824. Further, there may be more or fewer fields than those shown in Fig. 8, as represented by ellipses 828.

[0106] A window identifier 804 can include any identifier (ID) that uniquely identifies the associated window in relation to other windows in the window user interface 700. The window identifier 804 can be a globally unique identifier (GUID), a numeric ID, an alphanumeric ID, or other type of identifier. In embodiments, the window identifier 804 can be one, two, or any number of digits based on the number of windows that can be opened. In alternative embodiments, the size of the window identifier 804 may change based on the number of windows opened. While the window is open, the window identifier 804 may be static and remain unchanged.

[0107] Dimensions 808 can include dimensions for the window. The dimensions 808 can limit the window to a portion or all of the composite display 720. Thus, the dimensions 808 can be a set of display coordinates for the window. For example, the window can include a first set of coordinates for a first corner and a second set of coordinates for a second corner. In other embodiments, the dimensions 808 can include one set of coordinates and a length and width, measured in pixels or other measure.

[0108] A window stack position identifier 812 can be any identifier that can identify the position in the window stack for the window. The window stack position identifier 812 can be a GUID, a numeric ID, an alphanumeric ID, or other type of identifier. Each window or desktop can include a window stack position identifier 812. For example, as shown in Fig. 7C, window 3 708, in a window stack, can have a window stack position identifier 812 of 1 identifying that window 708 is the first window in the window stack and the active window. Similarly, window 2 712 can have a window stack position identifier 812 of 2 representing that window 712 is the

second window in the window stack. Window 1 704 can also have a window stack position identifier 812 of 3 representing that window 704 is the third window in the window stack.

[0109] A display identifier 816 can identify that the window or desktop is associated with either the first display 110 or the second display 114. While this display identifier 816 may not be needed for a dual user interface 700 system, as shown in Fig. 7A, the display identifier 816 can indicate whether a window in the serial user interface 700 is displayed on a particular display. Thus, window 3 708 may have a display identifier 816 of 2 in Fig. 7D. Window 1 704 can have a display identifier 816 of 1 for the first display 110. However, in alternative embodiments, the window may have two display indicators 816 that represent that the window is displayed on both of the displays 110, 114 or may have a single identifier identifying the composite display 720.

[0110] Similar to the display identifier 816, an active indicator 820 may not be needed with the dual user interface 700 system. In the device 100, the active indicator 820 can indicate which window(s) in the user interface 700 is being displayed. Thus, window 1 704 may have an active indicator 820 in Fig. 7D but may not have an active indicator in Fig. 7E.

[0111] An embodiment of a method 900 for creating a window user interface 700 is shown in Fig. 9. While a general order for the steps of the method 900 is shown in Fig. 9. Generally, the method 900 starts with a start operation 904 and ends with an end operation 924. The method 900 can include more or fewer steps or can arrange the order of the steps differently than those shown in Fig. 9. The method 900 can be executed as a set of computer-executable instructions executed by a computer system and encoded or stored on a computer readable medium. Hereinafrer, the method 900 shall be explained with reference to the systems, components, modules, software, data structures, user interfaces, etc. described in conjunction with Figs. 1-8.

[0112] A multi-screen device 100 can receive a movement of a window or other user interface event (such as the closure of a window, opening of a new window, revealing a desktop, etc., in step 908. While other user interface events are possible, the process 900 will be explained using a window movement as an example. One skilled in the art will understand how to apply the process 900 to other types of user interface events. In embodiments, the multi-screen device 100 can receive a window movement by receiving an input from the touch sensitive display 110 or 114, the configurable area 112 or 116, a gesture capture region 120 or 124, or some other hardware sensor operable to receive user interface inputs. The display controller 544 may send

the input signal to the processor 204 for processing. The processor 204 can interpret the input signal as a gesture, described in conjunction with Figs. 4A-4F, or other user input that moves an active or inactive window or moves a desktop. Upon receiving the gesture, the processor 204 can move the window. For example, window 3 708 can be moved onto the top of the first touch sensitive display 110, which makes window 2 712 active, as shown in Figs. 7E through 7G.

[0113] In embodiments, the processor 204 places the user interface interaction in the buffer 556 to be acted upon by the Input Management Module 536. Further, the processor 204 activates the inactive application, *i.e.*, activates the application from a suspended or sleep state. The application management class 540 may bind the application 564 to current display parameters. The device state module 574 of the Multi-Display Management class 524 may then determine how to rearrange the application in the user interface 700. Further, the device state module 574 may time the movement of a window. For example, before inactivating or activating a window. the device state module 574 may execute a timer. Only afrer the move is complete and the timer has expired will the device state module 574 activate or inactivate a window. The timer ensures that, if the user is flipping through multiple windows, then a window will not become active or other windows will not be put to sleep until the rapid movement of several windows ceases. the timer saves processor resources in not having to render user interfaces that will only be covered shortly thereafrer.

[0114] The device state module 574 can first determine if the moved window has changed position in the user interface 700, in step 912. In other words, the device state module 574 determines if to what position on the user interface, e.g., which display or to what position in a window stack, the window has been moved. Further, the device state module 574 may determine whether the movement of the window caused some other window to be covered in the user interface 700, in step 916. For example, the movement of window 3 708, show in Figs. 7E through 7G has changed the position of window 3 708 to the first touch sensitive display 110, has covered window 1 704, and has uncovered window 2 712. The device state module 574 determines these types of effects based on a window movement or other user interface event (e.g., opening a new window, closing a window, revealing the desktop, etc.).

[0115] Further, the display configuration module 568 places the hidden windows into a sleep state and activates uncovered windows, in step 920. Thus, the display configuration module 568 can change the active indicator 820 for the windows that are uncovered or covered in the user

interface 700. For example, window 1 704, which is covered in the embodiment shown in Fig. 7F, will have the active indicator eliminated to put the window to sleep. In the inactive or sleep state, window 1 stops receiving user interface inputs and stops rendering a user interface window. Further, window 2 712 has its active indicator set to acknowledge that window 2 712 is now active, receiving inputs, and rendering a window user interface. This sleep state information is passes to the Task Management Module 540 to instruct the Window Management Module 532 to render the new user interface 700 with the new active windows displayed and the other windows placed in the sleep state.

[0116] The exemplary systems and methods of this disclosure have been described in relation to a window stack associated with a dual screen device. However, to avoid unnecessarily obscuring the present disclosure, the preceding description omits a number of known structures and devices. This omission is not to be construed as a limitation of the scopes of the claims. Specific details are set forth to provide an understanding of the present disclosure. It should however be appreciated that the present disclosure may be practiced in a variety of ways beyond the specific detail set forth herein.

[0117] Furthermore, while the exemplary aspects, embodiments, and/or configurations illustrated herein show the various components of the system collocated, certain components of the system can be located remotely, at distant portions of a distributed network, such as a LAN and/or the Internet, or within a dedicated system. Thus, it should be appreciated, that the components of the system can be combined in to one or more devices, such as the software modules of Figs. 5A and 5B, or collocated on a particular node of a distributed network, such as an analog and/or digital telecommunications network, a packet-switch network, or a circuit-switched network. It will be appreciated from the preceding description, and for reasons of computational efficiency, that the components of the system can be arranged at any location within a distributed network of components without affecting the operation of the system. For example, the various components can be located in a switch such as a PBX and media server, gateway, in one or more communications devices, at one or more users' premises, or some combination thereof. Similarly, one or more functional portions of the system could be distributed between a telecommunications device(s) and an associated computing device.

[0118] Furthermore, it should be appreciated that the various links connecting the elements can be wired or wireless links, or any combination thereof, or any other known or later developed

element(s) that is capable of supplying and/or communicating data to and from the connected elements. These wired or wireless links can also be secure links and may be capable of communicating encrypted information. Transmission media used as links, for example, can be any suitable carrier for electrical signals, including coaxial cables, copper wire and fiber optics, and may take the form of acoustic or light waves, such as those generated during radio-wave and infra-red data communications.

[0119] Also, while the flowcharts have been discussed and illustrated in relation to a particular sequence of events, it should be appreciated that changes, additions, and omissions to this sequence can occur without materially affecting the operation of the disclosed embodiments, configuration, and aspects.

[0120] A number of variations and modifications of the disclosure can be used. It would be possible to provide for some features of the disclosure without providing others. For example in one alternative embodiment, the window stack 700 resembles a carousel or rolodex rather than a deck of cards. Thus, windows may circulate from one touch sensitive display 110 to another touch sensitive display 114. For example, window 6 724 may be push to the right and end up at the end of stack 764 behind window 716. If the stack continues to move to the right, eventually window 6 724 would appear on the second touch sensitive display 114 even though the window 724 was opened in the first touch sensitive display 110. These movements and changes in the stacks may be managed using the methods and logical data structures discussed above. In another alternative embodiment, there may be other arrangement for the window stack.

[0121] In yet another embodiment, the systems and methods of this disclosure can be implemented in conjunction with a special purpose computer, a programmed microprocessor or microcontroller and peripheral integrated circuit element(s), an ASIC or other integrated circuit, a digital signal processor, a hard-wired electronic or logic circuit such as discrete element circuit, a programmable logic device or gate array such as PLD, PLA, FPGA, PAL, special purpose computer, any comparable means, or the like. In general, any device(s) or means capable of implementing the methodology illustrated herein can be used to implement the various aspects of this disclosure. Exemplary hardware that can be used for the disclosed embodiments, configurations and aspects includes computers, handheld devices, telephones (e.g., cellular, Internet enabled, digital, analog, hybrids, and others), and other hardware known in the art. Some of these devices include processors (e.g., a single or multiple microprocessors), memory,

nonvolatile storage, input devices, and output devices. Furthermore, alternative software implementations including, but not limited to, distributed processing or component/object distributed processing, parallel processing, or virtual machine processing can also be constructed to implement the methods described herein.

[0122] In yet another embodiment, the disclosed methods may be readily implemented in conjunction with software using object or object-oriented software development environments that provide portable source code that can be used on a variety of computer or workstation platforms. Alternatively, the disclosed system may be implemented partially or fully in hardware using standard logic circuits or VLSI design. Whether software or hardware is used to implement the systems in accordance with this disclosure is dependent on the speed and/or efficiency requirements of the system, the particular function, and the particular software or hardware systems or microprocessor or microcomputer systems being utilized.

[0123] In yet another embodiment, the disclosed methods may be partially implemented in software that can be stored on a storage medium, executed on programmed general-purpose computer with the cooperation of a controller and memory, a special purpose computer, a microprocessor, or the like. In these instances, the systems and methods of this disclosure can be implemented as program embedded on personal computer such as an applet, JAVA® or CGI script, as a resource residing on a server or computer workstation, as a routine embedded in a dedicated measurement system, system component, or the like. The system can also be implemented by physically incorporating the system and/or method into a software and/or hardware system.

[0124] Although the present disclosure describes components and functions implemented in the aspects, embodiments, and/or configurations with reference to particular standards and protocols, the aspects, embodiments, and/or configurations are not limited to such standards and protocols. Other similar standards and protocols not mentioned herein are in existence and are considered to be included in the present disclosure. Moreover, the standards and protocols mentioned herein and other similar standards and protocols not mentioned herein are periodically superseded by faster or more effective equivalents having essentially the same functions. Such replacement standards and protocols having the same functions are considered equivalents included in the present disclosure.

[0125] The present disclosure, in various aspects, embodiments, and/or configurations, includes components, methods, processes, systems and/or apparatus substantially as depicted and described herein, including various aspects, embodiments, configurations embodiments, subcombinations, and/or subsets thereof. Those of skill in the art will understand how to make and use the disclosed aspects, embodiments, and/or configurations after understanding the present disclosure. The present disclosure, in various aspects, embodiments, and/or configurations, includes providing devices and processes in the absence of items not depicted and/or described herein or in various aspects, embodiments, and/or configurations hereof, including in the absence of such items as may have been used in previous devices or processes, e.g., for improving performance, achieving ease and\or reducing cost of implementation.

[0126] The foregoing discussion has been presented for purposes of illustration and description. The foregoing is not intended to limit the disclosure to the form or forms disclosed herein. In the foregoing Detailed Description for example, various features of the disclosure are grouped together in one or more aspects, embodiments, and/or configurations for the purpose of streamlining the disclosure. The features of the aspects, embodiments, and/or configurations of the disclosure may be combined in alternate aspects, embodiments, and/or configurations other than those discussed above. This method of disclosure is not to be interpreted as reflecting an intention that the claims require more features than are expressly recited in each claim. Rather, as the following claims reflect, inventive aspects lie in less than all features of a single foregoing disclosed aspect, embodiment, and/or configuration. Thus, the following claims are hereby incorporated into this Detailed Description, with each claim standing on its own as a separate preferred embodiment of the disclosure.

[0127] Moreover, though the description has included description of one or more aspects, embodiments, and/or configurations and certain variations and modifications, other variations, combinations, and modifications are within the scope of the disclosure, e.g., as may be within the skill and knowledge of those in the art, after understanding the present disclosure. It is intended to obtain rights which include alternative aspects, embodiments, and/or configurations to the extent permitted, including alternate, interchangeable and/or equivalent structures, functions, ranges or steps to those claimed, whether or not such alternate, interchangeable and/or equivalent structures, functions, ranges or steps are disclosed herein, and without intending to publicly dedicate any patentable subject matter.

What is claimed is:

1. A computer readable medium, having stored thereon, computer-executable instructions executable by a processor, the computer-executable instructions causing the processor to execute a method for changing a user interface for a device, the computer-executable instructions comprising:

instructions to receive an user interface event;

instructions to determine if a window is covered by the user interface event; and

if a window is covered by the user interface event, instructions to place any covered window into an inactive state.

2. The computer readable medium as defined in claim 1, further comprising:

instructions to determine if a window is uncovered by the user interface event; and

if a window is uncovered by the user interface event, instructions to place any uncovered window into an active state.

3. The computer readable medium as defined in claim 2, wherein the user interface event is a movement of a window.

4. The computer readable medium as defined in claim 3, wherein the new position in the window stack can be associated with at least one of two displays of the dual screen device.

5. The computer readable medium as defined in claim 1, further comprising:

instructions to execute a timer; and

instructions to wait until the timer expires before placing a window into an active or inactive state.

6. The computer readable medium as defined in claim 5, wherein two or more user interface events occur before the timer expires.

7. The computer readable medium as defined in claim 5, further comprising altering a logical data structure in response to placing a window in an inactive or active state.

8. The computer readable medium as defined in claim 7, wherein the logical data structure comprises one or more of:

a window identifier adapted to identify the window in relation to other windows in the user interface;

a display identifier adapted to identify which of one or more displays of a multi-screen screen device the window is associated; and

an active indicator adapted to identify whether the window is active or inactive.

9. The computer readable medium as defined in claim 8, wherein the active indicator is set when a window is uncovered.

10. The computer readable medium as defined in claim 9, wherein the active indicator is eliminated when a window is covered.

11. A device comprising:

a user interface comprising at least two displays disposed on at least one screen;

a memory;

a processor in communication with the memory and each of the two displays, the processor operable to:

receive an movement of a first window in the user interface, wherein the movement of the first window covers a second window;

determine that the second window was covered by the first window; and

place the second window into an inactive state.

12. The computer readable medium as defined in claim 11, wherein the first window is moved from a second display to a first display.

13. The computer readable medium as defined in claim 12, wherein the second window was displayed and active on the first display before the first window was moved.

14. The computer readable medium as defined in claim 13, wherein the processor is further operable to:

determine that a third window was uncovered by the movement of the first window; and

place the third window into an active state.

15. The computer readable medium as defined in claim 14, wherein, after the first window is moved, the third window is displayed and active on the second display.

16. A method for changing a user interface, the method comprising:

receiving an user interface event, wherein the user event causes at least one of covers a first window or uncovers a second window;

determining that the first window is covered or that the second window is uncovered; and

placing the first window into an inactive state or the second window into an active state.

17. The method defined in claim 16, further comprising, in response to placing the first window into the inactive state or placing the second window into an active state, changing a logic data structure associated with one of the first window or the second window.

18. The method defined in claim 16, wherein the logic data structure comprises:

a window identifier adapted to identify the window in relation to other windows in the user interface;

a display identifier adapted to identify which of one or more displays of a multi-screen screen device the window is associated; and

an active indicator adapted to identify whether the window is active or inactive.

19. The method defined in claim 16, wherein the user interface includes at least one other inactive window.

20. The method defined in claim 16, wherein the user event is one of a movement of a window, an opening of a window, revealing a desktop, or closing a window.

ABSTRACT

Systems and methods are provides for changing a user interface for a multi-screen device. The user interface can change based on the movement of a window. The system can receive a user interface event that modifies the display of windows in the user interface. Upon receiving the user interface event, the system determines if a window has been covered or uncovered. If a window has been covered, the window is placed in a sleep state. If a window is uncovered, the window is activated from a sleep state. A sleep state is a window state where an application associated with the window does not receive user interface inputs and/or does not render the window.

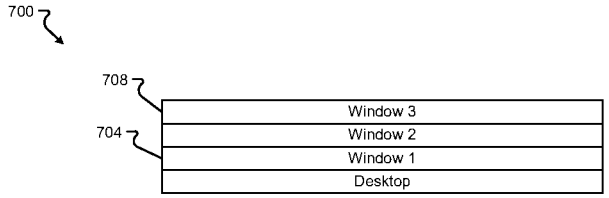


FIG. 7C

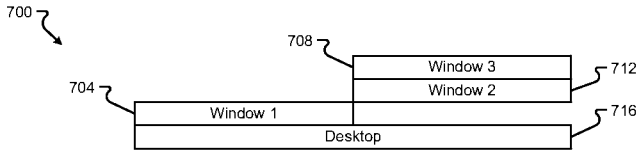


FIG. 7D

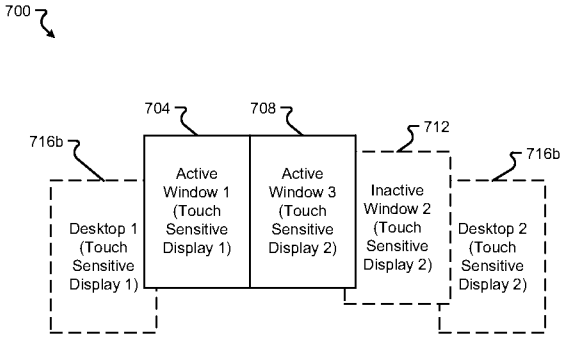


FIG. 7A

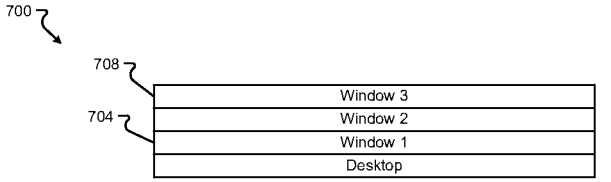


FIG. 7E

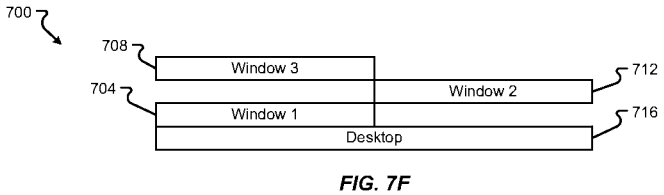


FIG. 7F

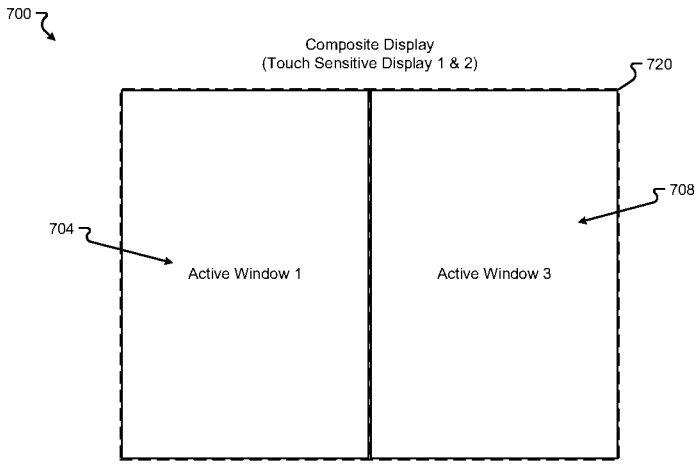


FIG. 7B

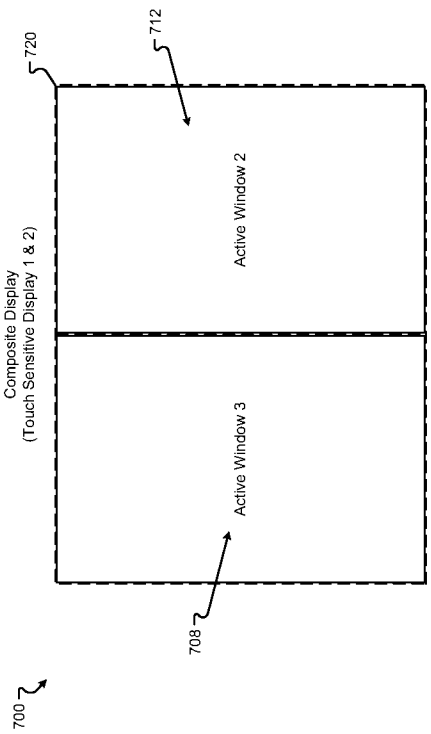


FIG. 7G

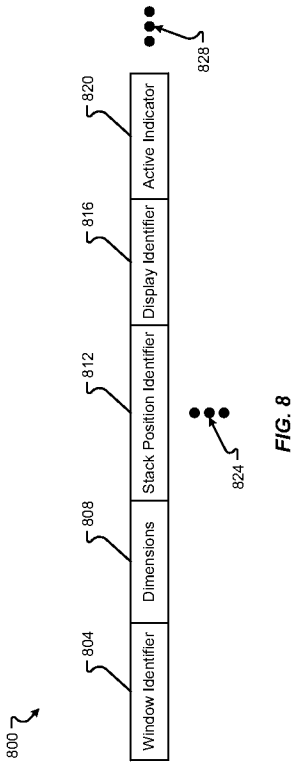


FIG. 8

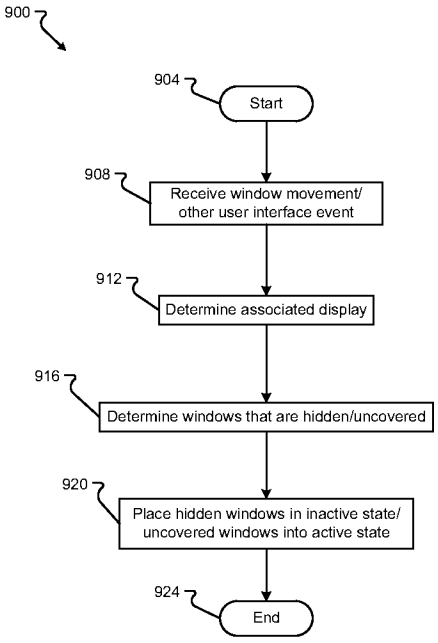


FIG. 9

METHOD AND APPARATUS FOR SHOWING STORED WINDOW DISPLAY

[0100] The device 100 manages desktops and/or windows within a user interface 700, as shown in Figs. 7A through 7G. A user interface 700 is a logical arrangement of active and inactive windows for a multi-screen device. For example, the user interface 700 may be logically similar to a deck of cards or a film strip, where one or more windows or desktops are arranged in order, as shown in Fig. 7A. An active window is a window that is currently being displayed on at least one of the touch sensitive displays 110, 114. For example, window 1 704 and window 3 708 are active windows and are displayed on touch sensitive displays 110 and 114, respectively. An inactive window is a window that was opened and displayed but is now "behind" an active window and not being displayed. In embodiments, an inactive window may be for an application that is suspended in a "sleep state", and thus, the window is not displaying active content. For example, window 2 712 is an inactive window in the sleep state.

[0101] A user interface 700 may have various arrangements or organizational structures. In the embodiment shown in Fig. 7A, the device 100 includes a set of windows/display objects or windows stack associated with a first touch sensitive display 110 and a second set of windows/display objects or windows stack associated with a second touch sensitive display 114. Thus, each touch sensitive display 110, 114 can have an associated set of windows/display objects. The two sets of windows/display objects may have different numbers of windows and may be arranged differently. Further, the two sets of windows/display objects can also be identified differently and managed separately. Thus, the first set of windows/display objects can be arranged in order from a first window to a last window. Likewise, the second set of windows/display objects can be arranged from a first window to a last window, but may have more or fewer windows. A logical data structure for managing the sets of windows/display objects may be as described in conjunction with Fig. 8.

[0102] When a window is moved, the newly moved window is repositioned within the user interface 700. Where and how the window is positioned within the user interface 700 can be a function of the orientation of the device 100, the context of what programs, functions, software, etc. are being executed on the device 100, how the user interface 700 is arranged when a new window is opened, when a window is moved, etc. To change the window position in the user interface 700, the window is moved by a user and information associated with the window can change. With this information, a logical data structure for the window can be changed and

stored. When user interface or other events or tasks change the arrangement of windows, the user interface 700 can be changed to reflect the change in arrangement. It should be noted that these same concepts described above can be used to manage the one or more desktops for the device 100.

[0103] For example, an arrangement of windows in a user interface can be shown in "profile" views in Figs. 7B through 7G. A top view is shown in Figs. 7B and 7G. Side views are shown in Figs. 7C-7F. Here, the logical arrangement of the user interface 700 is shown where, the active windows 704 and 708 are shown in Fig. 7B. The windows 704 and 708 can be the top most windows in portions of a composite display 720, which represents the total area of all the touch sensitive displays 110,114. As shown in Figs. 7C and 7D, window 1 704 and window 3 708 are at the top of the set of windows/display objects. The windows "beneath" the top windows, window 1 704 and window 3 708, are not active. Rather, window 2 712, which is not displayed, is inactive or in a sleep state. In a sleep state, the window 2 712 does not receive inputs. Further, the window need not generate a user interface window. As such, by setting the window 2 712 inactive, the processor saves on processor cycles and required code execution. When a window is "awakened", it is brought back to an active status, begins receiving inputs, and renders a window user interface. As shown in Figs. 7C and 7D, window 2 712 is below window 3 708 and not displayed, which changes in the embodiment shown in Figs. 7E and 7F.

[0104] In Figs. 7E through 7G, window 3 708 is moved from the second touch sensitive display 114 onto a first touch sensitive display 110 by gesture or other input to the device 110. In moving window 3 704, window 1 704 is covered and becomes inactive. Window 2 712 is exposed and becomes active. When a window 1 704 becomes inactive, a bitmap of the last view of the window is stored. The bitmap can be used when windows transfer onto the active display(s). Thus, the window appears to be active when transitioning but is not actually rendering a user interface until the transition is complete, and the window becomes active. Window 2 712 appears as the bitmap until window 3 708 has completed the move to the first touch sensitive display 110, and window 2 becomes active, as shown on Fig. 7G.

[0105] A logical data structure 800 for managing the arrangement of windows or desktops in a window user interface 700 is shown in Fig. 8. The logical data structure 800 can be any data structure used to store data whether an object, record, file, etc. The logical data structure 800 can be stored in any type of database or data storage system, regardless of protocol or standard. In

embodiments, the logical data structure 800 includes one or more portions, fields, attributes, etc. that store data in a logical arrangement that allows for easy storage and retrieval of the information. Hereinafter, these one or more portions, fields, attributes, etc. shall be described simply as fields. The fields can store data for a window identifier 804, a dimension 808, a window stack position identifier 812, a display identifier 816, and/or an active indicator 820. Each window in a window user interface 700 can have an associated logical data structure 800. While only a single logical data structure 800 is shown in Fig. 8, there may be more or fewer logical data structures 800 used with a window user interface 700 (based on the number of windows or desktops in the user interface 700), as represented by ellipses 824. Further, there may be more or fewer fields than those shown in Fig. 8, as represented by ellipses 828.

[0106] A window identifier 804 can include any identifier (ID) that uniquely identifies the associated window in relation to other windows in the window user interface 700. The window identifier 804 can be a globally unique identifier (GUID), a numeric ID, an alphanumeric ID, or other type of identifier. In embodiments, the window identifier 804 can be one, two, or any number of digits based on the number of windows that can be opened. In alternative embodiments, the size of the window identifier 804 may change based on the number of windows opened. While the window is open, the window identifier 804 may be static and remain unchanged.

[0107] Dimensions 808 can include dimensions for the window. The dimensions 808 can limit the window to a portion or all of the composite display 720. Thus, the dimensions 808 can be a set of display coordinates for the window. For example, the window can include a first set of coordinates for a first corner and a second set of coordinates for a second corner. In other embodiments, the dimensions 808 can include one set of coordinates and a length and width, measured in pixels or other measure.

[0108] A window stack position identifier 812 can be any identifier that can identify the position in the window stack for the window. The window stack position identifier 812 can be a GUID, a numeric ID, an alphanumeric ID, or other type of identifier. Each window or desktop can include a window stack position identifier 812. For example, as shown in Fig. 7C, window 3 708, in a window stack, can have a window stack position identifier 812 of 1 identifying that window 708 is the first window in the window stack and the active window. Similarly, window 2 712 can have a window stack position identifier 812 of 2 representing that window 712 is the

second window in the window stack. Window 1 704 can also have a window stack position identifier 812 of 3 representing that window 704 is the third window in the window stack.

[0109] A display identifier 816 can identify that the window or desktop is associated with either the first display 110 or the second display 114. While this display identifier 816 may not be needed for a dual user interface 700 system, as shown in Fig. 7A, the display identifier 816 can indicate whether a window in the serial user interface 700 is displayed on a particular display. Thus, window 3 708 may have a display identifier 816 of 2 in Fig. 7D. Window 1 704 can have a display identifier 816 of 1 for the first display 110. However, in alternative embodiments, the window may have two display indicators 816 that represent that the window is displayed on both of the displays 110, 114 or may have a single identifier identifying the composite display 720.

[0110] Similar to the display identifier 816, an active indicator 820 may not be needed with the dual user interface 700 system. In the device 100, the active indicator 820 can indicate which window(s) in the user interface 700 is being displayed. Thus, window 1 704 may have an active indicator 820 in Fig. 7D but may not have an active indicator in Fig. 7E.

[0111] An embodiment of a method 900 for creating a window user interface 700 is shown in Fig. 9. While a general order for the steps of the method 900 is shown in Fig. 9. Generally, the method 900 starts with a start operation 904 and ends with an end operation 924. The method 900 can include more or fewer steps or can arrange the order of the steps differently than those shown in Fig. 9. The method 900 can be executed as a set of computer-executable instructions executed by a computer system and encoded or stored on a computer readable medium. Hereinafter, the method 900 shall be explained with reference to the systems, components, modules, software, data structures, user interfaces, etc. described in conjunction with Figs. 1-8.

[0112] A multi-screen device 100 can receive a movement of a window or other user interface event (such as the closure of a window, opening of a new window, revealing a desktop, etc.), in step 908. While other user interface events are possible, the process 900 will be explained using a window movement as an example. One skilled in the art will understand how to apply the process 900 to other types of user interface events. In embodiments, the multi-screen device 100 can receive a window movement by receiving an input from the touch sensitive display 110 or 114, the configurable area 112 or 116, a gesture capture region 120 or 124, or some other hardware sensor operable to receive user interface inputs. The display controller 544 may send

the input signal to the processor 204 for processing. The processor 204 can interpret the input signal as a gesture, described in conjunction with Figs. 4A-4F, or other user input that moves an active or inactive window or moves a desktop. Upon receiving the gesture, the processor 204 can move the window. For example, window 3 708 can be moved onto the top of the first touch sensitive display 110, which makes window 2 712 active, as shown in Figs. 7E through 7G.

[0113] In embodiments, the processor 204 places the user interface interaction in the buffer 556 to be acted upon by the Input Management Module 536. Further, the processor 204 activates the inactive application, *i.e.*, activates the application ftom a suspended or sleep state. The application management class 540 may bind the application 564 to current display parameters. The device state module 574 of the Multi-Display Management class 524 may then determine how to rearrange the application in the user interface 700. Further, the device state module 574 may time the movement of a window. For example, before inactivating or activating a window. the device state module 574 may execute a timer. Only after the move is complete and the timer has expired will the device state module 574 activate or inactivate a window. The timer ensures that, if the user is flipping through multiple windows, then a window will not become active or other windows will not be put to sleep until the rapid movement of several windows ceases. the timer saves processor resources in not having to render user interfaces that will only be covered shortly thereafter.

[0114] The device state module 574 can first determine if the moved window has changed position in the user interface 700, in step 912. In other words, the device state module 574 determines to what position on the user interface, e.g., which display or to what position in a window stack, the window has been moved. Further, the device state module 574 may determine whether the movement of the window caused some other window to be covered in the user interface 700, in step 916. For example, the movement of window 3 708, show in Figs. 7E through 7G has changed the position of window 3 708 to the first touch sensitive display 110, has covered window 1 704, and has uncovered window 2 712. The device state module 574 determines these types of effects based on a window movement or other user interface event (e.g., opening a new window, closing a window, revealing the desktop, etc.).

[0115] Further, the display configuration module 568 places the hidden windows into a sleep state and activates uncovered windows, in step 920. Thus, the display configuration module 568 can change the active indicator 820 for the windows that are uncovered or covered in the user

interface 700. For example, window 1 704, which is covered in the embodiment shown in Fig. 7F, will have the active indicator eliminated to put the window to sleep. In the inactive or sleep state, window 1 stops receiving user interface inputs and stops rendering a user interface window. Further, window 2 712 has its active indicator set to acknowledge that window 2 712 is now active, receiving inputs, and rendering a window user interface. This sleep state information is passed to the Task Management Module 540 to instruct the Window Management Module 532 to render the new user interface 700 with the new active windows displayed and the other windows placed in the sleep state.

[0116] Fig. 10A depicts displayed windows 704 and 712, and other windows in a set or stack of windows 1000, maintained as part of a user interface 700. In this example, a first window 704 is presented by a first screen 104 of the device 100, in an active and displayed state. The second window 712 is presented by the second screen 108 of the device in an inactive and displayed state, also known as a sleep state. Other windows within the set of windows 1000 maintained by the user interface 700 include a third window 708, a fourth window 710, a first desktop 716a, and a second desktop 716b, and are maintained in an active and not displayed state. In accordance with further embodiments, the set of windows 1000 can be divided into or can comprise multiple sets of windows. For example, the first window 704, the fourth window 710, and the first desktop 716a can be included in a first set of windows 1000a that are generally associated with the first screen 104, at least at the point in time illustrated in the figure. Continuing this example, the second window 712, third window 708, and second desktop 716b can be included in a second set of windows 1000b, that is generally associated with the second screen 108, at least at the time of this example.

[0117] Fig. 10B illustrates the user interface 700 of Fig. 10A, after a user input affecting the window presented by the second screen 108 of the device 100 has been received. In accordance with embodiments of the present invention, the user input can be in the form of an off-screen gesture or other windowing management input. In the example of Fig. 10B, the received user input has replaced the second window 712 in the second screen 108 with the third window 708. Moreover, Fig. 10B illustrates the state of the user interface 700 immediately after the user input is acted upon. In this state, the third window 708 is presented by the second screen 108 in an inactive and displayed state. More particularly, immediately following the receipt of the user input resulting in the presentation of the third window 708, that third window 708 remains in an

inactive or sleep state. In addition, the second window 712 is maintained in the set of windows 1000 in an inactive and not displayed state. Although in the present example, the second window 712 was in an inactive and displayed state at the point in time in which the user input was received, even if the second window 712 had been in an active and displayed state at the time the user input was received, after acting on that user input, the second window 712 would be placed in an inactive or sleep state, and would be maintained in the set of windows 1000 in the inactive state. More particularly, embodiments of the present invention maintain windows that are not displayed in an inactive state, conserving processing resources. Moreover, where a window that was in an active and displayed state is replaced in response to received user input or other instructions, that window is placed in an inactive or sleep state at or about the time that the window ceases to be displayed.

[0118] Also at the time input is received to display a different window on a screen 104 or 108 of a device 100, a timer can be started. Until that timer expires, the newly displayed window (e.g., the third window 708 in this example) remains in an inactive or sleep state. Therefore, if additional input replacing the newly displayed window (e.g., the third window 708) with another window (e.g., the second desktop 716b or the second window 712), the window thus replaced (e.g., the third window 708) is not activated or otherwise taken out of the sleep state before it is placed back into an inactive and not displayed state.

[0119] Fig. 10C depicts the user interface 700 as shown in Fig. 10B, but after the expiration of a predetermined period of time. For example, the period of time can correspond to the expiration of a timer set when a newly displayed window (e.g., the third window 708) is presented in response to user input. Accordingly, in Fig. 10C, after this predetermined period of time has elapsed, the third window 708 is activated, thereby placing that window in an active and displayed state. In the active and displayed state, an application or objects presented by the window 708 can be executed and/or manipulated normally.

[0120] In accordance with embodiments implemented in connection with a device 100 having first 104 and second 108 screens, user input to control the window or other information displayed by the device 100 can be performed with respect to either or both screens 104 and 108. Therefore, although certain scenarios described up to this point as examples of the operation of a device 100 and in particular a user interface 700 implemented thereby have involved changes to the second screen 108, similar operations can be performed with respect to the first screen 104.

Moreover, separate timers can be implemented in connection with the displays of the screens 104 and 108. For example, if a user input is received to discontinue the presentation of the first window 704, and to instead display the fourth window 710 in the first screen 104, and such input is received while the user interface 700 is in the state illustrated in Fig. 10B, the resulting state immediately following the receipt of such input is depicted in Fig. 10D. In Fig. 10D, the first screen presents the fourth window 710 in an inactive and displayed state, while the first window 704 is maintained in the stack in an inactive and not displayed state. Moreover, in this example, the third window 708 continues in a displayed and inactive state. Accordingly, Fig. 10D depicts a situation in which both of the screens 104 and 108 present windows (here windows 710 and 708) that are simultaneously in an inactive and displayed state. In accordance with embodiments of the present invention, each timer that is started when a previously un-displayed window is presented applies the same predetermined period of time in connection with a determination as to whether to place a displayed window in an active state. Accordingly, following the example discussed in connection with Fig. 10D, provided no additional user input in the form of windowing management input affecting either of the displayed windows, the third window 708 will be restarted or placed in an active state prior to the time at which the fourth window 710 is restarted or placed in an active state.

[0121] In accordance with embodiments of the present invention, in an inactive or sleep state, any application associated with a window is not executed. Moreover, an image representing the window is stored. In accordance with embodiments of the present invention, the image depicting the window can comprise an image of that window as it appeared in a last active state. Accordingly, the image that is stored can comprise a screen shot of the window at the time that it was placed in the inactive state. The image representing an inactive window that is maintained in the window stack can therefore provide a representation of the window with little or no delay if that window is placed in a displayed and inactive state. As a further example, the image can comprise an icon representing an application or information related to the window, a pro forma representation of a display screen or other output associated with the window, or any other graphical representation. Moreover, an image representing an inactive window, whether a screen shot of an actual operational state or otherwise, may be stored as a bitmap image in memory 508. Furthermore, images of windows can be stored as part of a logical data structure 800. In accordance with still other embodiments, a window in an inactive and displayed state can be

presented as a dimmed image, or can otherwise include a graphic indication to the user that the window is in an inactive or sleep state.

[0122] Fig. 11 illustrates aspects of a method for selectively showing a stored window or display object in accordance with embodiments of the present invention. Initially, at step 1104, a window stack or set of windows is provided. In the window stack, information related to active and inactive windows associated with the user interface 700 is maintained. More particularly, for each such window, various information is maintained, including but not limited to whether the window is in an active and displayed state, an inactive and displayed state, or a not displayed and inactive state. In addition, for each window, an image comprising a screen shot or other representation of the window can be maintained. At step 1108, a determination is made as to whether a windowing management input is received from a user. Until a windowing management input from a user or an instruction generated through operation of application programming or otherwise is received, the process may idle at step 1108.

[0123] Upon the receipt of windowing management input from a user, or instructions generated through the execution of application programming that signals or includes a signal to uncover a window that is not displayed, the window that is to be uncovered is presented (step 1112). In addition, at or about the time the window selected to be uncovered is presented, the window previously presented by the screen 104 or 108 then occupied by the window selected to be covered is put in an inactive and not displayed state in the set of windows (step 1116). The information presented by the screen or screens 104, 108 of the device 100 is thereby updated. Upon or after the presentation of the window selected to be uncovered by a screen 104 or 108 of the device 100, a timer, for example implemented by application programming, can be started (step 1120). While the timer is active, the window that was newly selected continues to be presented in an inactive and displayed state. Also while the timer is active, the user interface 700 can continue to look for user input, for example in the form of windowing management input (step 1122). If input is received instructing the user interface 700 to uncover yet another window and cover the window currently presented in an inactive and displayed state, the process can return to steps 1112 and 1116, to take action in response to those instructions. In that case, since the window being covered is already inactive, the display of that window is simply discontinued, while being maintained in the set of windows associated with the applicable screen 104 or 108. If such input is not received, the process can continue.

[0124] At step 1124, a determination is made as to whether the timer has expired. If the timer has expired, the newly uncovered window is placed in an active and displayed state (step 1128). If the timer has not yet expired, the process can return to step 1122, to look for additional windowing management input from the user.

[0125] Where the timer has expired and the presented window has been placed in an active and displayed state, a determination can be made as to whether the device has been powered off or the method for controlling a display of a window on a device 100 has otherwise been discontinued (step 1132). If operation of the device is continued, the process can return to step 1108. Otherwise, the process may end.

[0126] Although the method has, at least in some examples, been described in connection with a single screen 104 or 108 of a device 100, it should be appreciated by one of skill in the art after consideration of the present disclosure that embodiments of the present invention can be implemented in connection with two or more screens of a device 100. For example, by maintaining multiple sets of windows and running processes for managing the display of windows from the different sets of windows in parallel, the presentation of multiple displays can be managed.

[0127] Various other exemplary systems and methods of this disclosure have been described in relation to the management of windows in a window stack associated with a multiple screen device. However, such other systems and methods can be applied to operation of a single screen device, or alternatively to a device with more than two screens. In addition, embodiments of the present invention can be implemented in connection with a device 100 presenting or implementing multiple virtual screens.

[0128] Although embodiments of systems and methods have been described in connection with various structures and attributes, in order to avoid unnecessarily obscuring the present disclosure, the preceding description omits a number of known structures and devices. This omission is not to be construed as a limitation of the scopes of the claims. Specific details are set forth to provide an understanding of the present disclosure. It should however be appreciated that the present disclosure may be practiced in a variety of ways beyond the specific detail set forth herein.

[0129] Furthermore, while the exemplary aspects, embodiments, and/or configurations illustrated herein show the various components of the system collocated, certain components of

the system can be located remotely, at distant portions of a distributed network, such as a LAN and/or the Internet, or within a dedicated system. Thus, it should be appreciated, that the components of the system can be combined in to one or more devices, such as the software modules of Figs. 5A and 5B, or collocated on a particular node of a distributed network, such as an analog and/or digital telecommunications network, a packet-switch network, or a circuit-switched network. It will be appreciated from the preceding description, and for reasons of computational efficiency, that the components of the system can be arranged at any location within a distributed network of components without affecting the operation of the system. For example, the various components can be located in a switch such as a PBX and media server, gateway, in one or more communications devices, at one or more users' premises, or some combination thereof. Similarly, one or more functional portions of the system could be distributed between a telecommunications device(s) and an associated computing device.

[0130] Furthermore, it should be appreciated that the various links connecting the elements can be wired or wireless links, or any combination thereof, or any other known or later developed element(s) that is capable of supplying and/or communicating data to and from the connected elements. These wired or wireless links can also be secure links and may be capable of communicating encrypted information. Transmission media used as links, for example, can be any suitable carrier for electrical signals, including coaxial cables, copper wire and fiber optics, and may take the form of acoustic or light waves, such as those generated during radio-wave and infra-red data communications.

[0131] Also, while the flowcharts have been discussed and illustrated in relation to a particular sequence of events, it should be appreciated that changes, additions, and omissions to this sequence can occur without materially affecting the operation of the disclosed embodiments, configuration, and aspects.

[0132] A number of variations and modifications of the disclosure can be used. It would be possible to provide for some features of the disclosure without providing others. For example in one alternative embodiment, the window stack 1000 maintained in connection with the user interface 700 resembles a carousel or rolodex rather than a deck of cards. Thus, windows may circulate from one touch sensitive display 110 to another touch sensitive display 114. For example, window 6 724 may be pushed to the right and end up at the end of stack 764 behind window 716. If the stack continues to move to the right, eventually window 6 724 would appear

on the second touch sensitive display 114 even though the window 724 was opened in the first touch sensitive display 110. These movements and changes in the stacks may be managed using the methods and logical data structures discussed above. In another alternative embodiment, there may be other arrangement for the window stack.

[0133] In yet another embodiment, the systems and methods of this disclosure can be implemented in conjunction with a special purpose computer, a programmed microprocessor or microcontroller and peripheral integrated circuit element(s), an ASIC or other integrated circuit, a digital signal processor, a hard-wired electronic or logic circuit such as discrete element circuit, a programmable logic device or gate array such as PLD, PLA, FPGA, PAL, special purpose computer, any comparable means, or the like. In general, any device(s) or means capable of implementing the methodology illustrated herein can be used to implement the various aspects of this disclosure. Exemplary hardware that can be used for the disclosed embodiments, configurations and aspects includes computers, handheld devices, telephones (e.g., cellular, Internet enabled, digital, analog, hybrids, and others), and other hardware known in the art. Some of these devices include processors (e.g., a single or multiple microprocessors), memory, nonvolatile storage, input devices, and output devices. Furthermore, alternative software implementations including, but not limited to, distributed processing or component/object distributed processing, parallel processing, or virtual machine processing can also be constructed to implement the methods described herein.

[0134] In yet another embodiment, the disclosed methods may be readily implemented in conjunction with software using object or object-oriented software development environments that provide portable source code that can be used on a variety of computer or workstation platforms. Alternatively, the disclosed system may be implemented partially or fully in hardware using standard logic circuits or VLSI design. Whether software or hardware is used to implement the systems in accordance with this disclosure is dependent on the speed and/or efficiency requirements of the system, the particular function, and the particular software or hardware systems or microprocessor or microcomputer systems being utilized.

[0135] In yet another embodiment, the disclosed methods may be partially implemented in software that can be stored on a storage medium, executed on programmed general-purpose computer with the cooperation of a controller and memory, a special purpose computer, a microprocessor, or the like. In these instances, the systems and methods of this disclosure can be

implemented as program embedded on personal computer such as an applet, JAVA® or CGI script, as a resource residing on a server or computer workstation, as a routine embedded in a dedicated measurement system, system component, or the like. The system can also be implemented by physically incorporating the system and/or method into a software and/or hardware system.

[0136] Although the present disclosure describes components and functions implemented in the aspects, embodiments, and/or configurations with reference to particular standards and protocols, the aspects, embodiments, and/or configurations are not limited to such standards and protocols. Other similar standards and protocols not mentioned herein are in existence and are considered to be included in the present disclosure. Moreover, the standards and protocols mentioned herein and other similar standards and protocols not mentioned herein are periodically superseded by faster or more effective equivalents having essentially the same functions. Such replacement standards and protocols having the same functions are considered equivalents included in the present disclosure.

[0137] The present disclosure, in various aspects, embodiments, and/or configurations, includes components, methods, processes, systems and/or apparatus substantially as depicted and described herein, including various aspects, embodiments, configurations embodiments, subcombinations, and/or subsets thereof. Those of skill in the art will understand how to make and use the disclosed aspects, embodiments, and/or configurations after understanding the present disclosure. The present disclosure, in various aspects, embodiments, and/or configurations, includes providing devices and processes in the absence of items not depicted and/or described herein or in various aspects, embodiments, and/or configurations hereof, including in the absence of such items as may have been used in previous devices or processes, e.g., for improving performance, achieving ease and\or reducing cost of implementation.

[0138] The foregoing discussion has been presented for purposes of illustration and description. The foregoing is not intended to limit the disclosure to the form or forms disclosed herein. In the foregoing Detailed Description for example, various features of the disclosure are grouped together in one or more aspects, embodiments, and/or configurations for the purpose of streamlining the disclosure. The features of the aspects, embodiments, and/or configurations of the disclosure may be combined in alternate aspects, embodiments, and/or configurations other than those discussed above. This method of disclosure is not to be interpreted as reflecting an

intention that the claims require more features than are expressly recited in each claim. Rather, as the following claims reflect, inventive aspects lie in less than all features of a single foregoing disclosed aspect, embodiment, and/or configuration. Thus, the following claims are hereby incorporated into this Detailed Description, with each claim standing on its own as a separate preferred embodiment of the disclosure.

[0139] Moreover, though the description has included description of one or more aspects, embodiments, and/or configurations and certain variations and modifications, other variations, combinations, and modifications are within the scope of the disclosure, e.g., as may be within the skill and knowledge of those in the art, after understanding the present disclosure. It is intended to obtain rights which include alternative aspects, embodiments, and/or configurations to the extent permitted, including alternate, interchangeable and/or equivalent structures, functions, ranges or steps to those claimed, whether or not such alternate, interchangeable and/or equivalent structures, functions, ranges or steps are disclosed herein, and without intending to publicly dedicate any patentable subject matter.

What is claimed is:

1. A method for controlling a display of a window on a device, comprising:

providing a windows stack, wherein at least a first window in the windows stack is in an active and displayed state, wherein at least a second window in the windows stack is in an inactive and not displayed state, and wherein the second window is stored in memory as one of an image of the second window as it appeared in one of a last active state or an image representing the second window;

presenting using at least a first screen of the device the first window in the active and displayed of state;

receiving a first user input, wherein the first user input includes a signal to uncover the second window and cover the first window;

discontinuing the presentation of the first window;

presenting using the at least a first screen the second window, wherein the second window is presented in an inactive and displayed state.

2. The method of claim 1, wherein while displayed in the inactive state the second window is displayed using the at least a first screen as the one of an image of the second window as it appeared in a last active state or an image representing the second window.

3. The method of claim 2, further comprising:

putting the first window in an inactive and not displayed state, wherein the first window is stored in memory as one of an image of the first window as it appeared in one of a last active state when the user input including a signal to uncover the second window and cover the first window was received or an image representing the first window.

4. The method of claim 3, further comprising:

initiating operation of a timer at a time corresponding to the presenting the second window.

5. The method of claim 4, further comprising:

at an expiration of a predetermined time, as determined by the timer, placing the second window in an active and displayed state.

6. The method of claim 5, wherein prior to the expiration of the predetermined time, the second window remains in the inactive and displayed state.

7. The method of claim 3, wherein a third window in the windows stack is in an inactive and not displayed state, and wherein the third window is stored in memory as one of an image of the third window as it appeared in one of a last active state or an image representing the third window, the method further comprising:

receiving a second user input, wherein the second user input includes a signal to uncover the third window and cover the second window;

discontinuing the presentation of the second window;

presenting using the at least a first screen the third window, wherein the third window is presented in an inactive and displayed state.

8. The method of claim 7, wherein during the presentation of the second window between at least the receipt of the first user input and the receipt of the second user input the second window remains in the inactive state and displayed state.

9. The method of claim 1, wherein a third window in the windows stack is in an active and displayed state, the method further comprising:

presenting using a second screen of the device the third window in the active and displayed state.

10. The method of claim 9, wherein the third window remains in the active and displayed state during the discontinuation of the presentation of the first window and the presentation of the second window.

11. A device, comprising:

a first screen, wherein the first screen includes a first touch sensitive display;

memory;

a processor in communication with the first screen and the memory;

application programming stored in the memory and executed by the processor, wherein the application programming is operable to:

maintain at least a first set of windows, wherein the at least a first set of windows is stored in the memory, wherein the first set of windows includes a plurality of windows, wherein at least a first window in the first set of windows is in one of an active and displayed state or an inactive and displayed state, wherein at least a second window in the first set of windows is in an inactive and not displayed state, and wherein the second window is stored in memory as one of an image of the second window as it appeared in one of a last active state or an image representing the second window;

present the first window in the one of the active and displayed state or the inactive and displayed state using the first screen;

receive a first user input, wherein the first user input includes a signal to present the second window and discontinue the presentation of the first window;

in response to the first user input, discontinue the presentation of the first window and present the second window in an inactive and displayed state using the first screen.

12. The device of claim 11, wherein the second window is presented in the form of the image stored in memory for at least a first period of time.

13. The device of claim 11, wherein the application programming is further operable to continue to implement a timer, wherein after the expiration of the first period of time the second window is activated and is presented in an active and displayed state.

14. The device of claim 12, wherein the plurality of windows in the first set of windows includes a third window, wherein the application programming is further operable to:

receive a second user input, wherein the second user input includes a signal to present the third window and discontinue the presentation of the second window, wherein the second user input is received while the second window is presented in the inactive and displayed state;

in response to the second user input, discontinue the presentation of the second window and present the second window in an inactive and displayed state.

15. The device of claim 11, further comprising:

a second screen, wherein the second screen is in communication with the processor, wherein the application programming is further operable to maintain a second set of windows, wherein the second set of windows is stored in memory, wherein the second set of windows includes a third window, wherein the third window is in an active and displayed state, and wherein the third window is presented in the active and displayed state by the second screen for at least a portion of a time that the first window is displayed by the first screen and for at least a portion of a time that the second window is displayed by the first screen.

16. A computer readable medium having stored thereon computer-executable instructions, the computer executable instructions causing a processor to execute a method for controlling a user interface of a device, the computer-executable instructions comprising:

instructions to maintain a first set of windows, wherein at least a first window is maintained in the first set of windows as an active and displayed window, and wherein at least a second window is maintained in the first set of windows as an inactive and not displayed window;

instructions to present the first window on a first screen of the device in the active and displayed state;

instructions to receive a first user interface event, wherein the first user interface event directs the second window to be placed in a displayed state;

instructions to, in response to the first user interface event directing the second window to be placed in a displayed state, place the first window in an inactive and not displayed state and to present the second window on the first screen of the device in an inactive and displayed state.

17. The computer readable medium of claim 16, wherein in the inactive and displayed state the second window is presented as an image of the window in a last active state.

18. The computer readable medium of claim 16, the computer executable instructions further comprising:

instructions to maintain a second set of windows, wherein at least a third window is maintained in the second set of windows as an active and displayed window, and wherein at least a fourth window is maintained in the second set of windows as an inactive and not displayed window;

instructions to present the third window on a second screen of the device in an active and displayed state;

instructions to receive a second user interface event, wherein the second user interface event directs the fourth window to be placed in a displayed state;

instructions to, in response to the second user interface event directing the fourth window to be placed in a displayed state, place the third window in an inactive and not displayed state

and to present the fourth window on the second screen of the device in an inactive and displayed state.

19. The computer readable medium of claim 18, wherein for at least a first period of time the second window is presented on the first screen of the device in the inactive and displayed state and the fourth window is presented on the second screen of the device in the inactive and displayed state.

20. The computer readable medium of claim 19, wherein first user interface event is received before the second user interface event, and wherein after the first period of time and for at least a second period of time the second window is presented in an active and displayed state and the fourth window is displayed in the inactive and displayed state.

ABSTRACT

Systems and methods are provided for changing a user interface for a multi-screen device. The user interface can change based on the movement of a window. The system can receive a user interface event that modifies the display of windows in the user interface. Upon receiving the user interface event, the system determines if a window has been covered or uncovered. If a window has been covered, the window is placed in a sleep state. If a window is uncovered, the window is activated from a sleep state. A sleep state is a window state where an application associated with the window does not receive user interface inputs and/or does not render the window. Moreover, in a sleep state an image representing the window is maintained in memory.

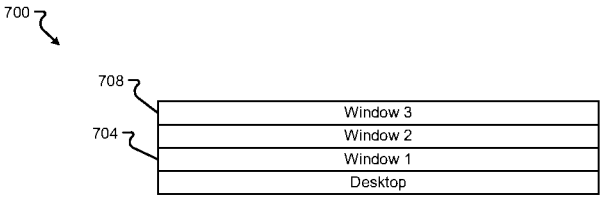


FIG. 7C

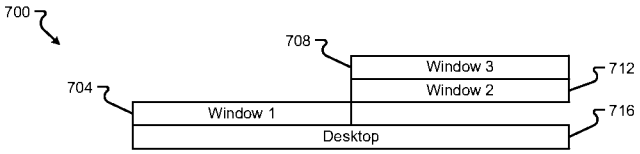


FIG. 7D

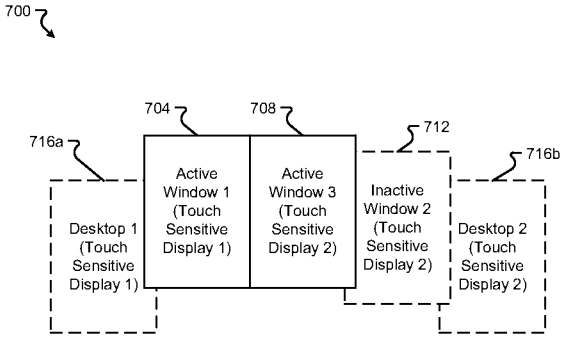


FIG. 7A

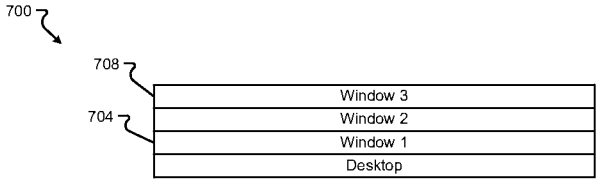


FIG. 7E

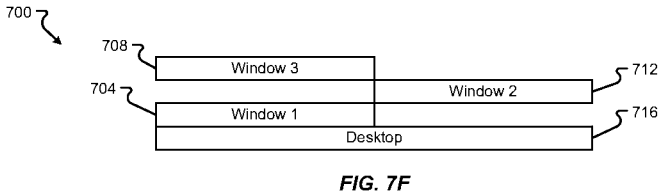


FIG. 7F

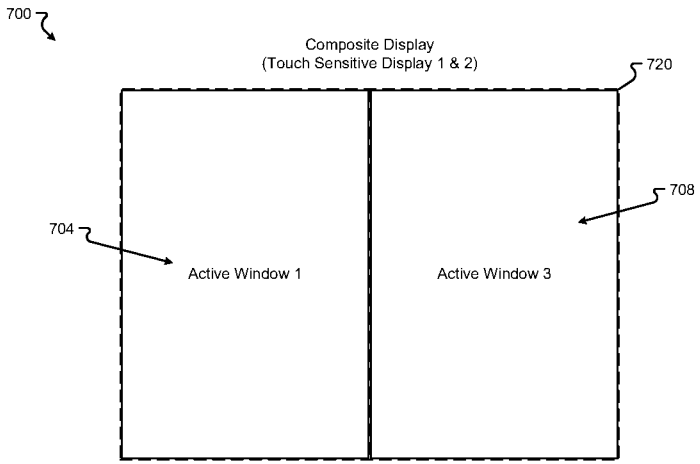


FIG. 7B

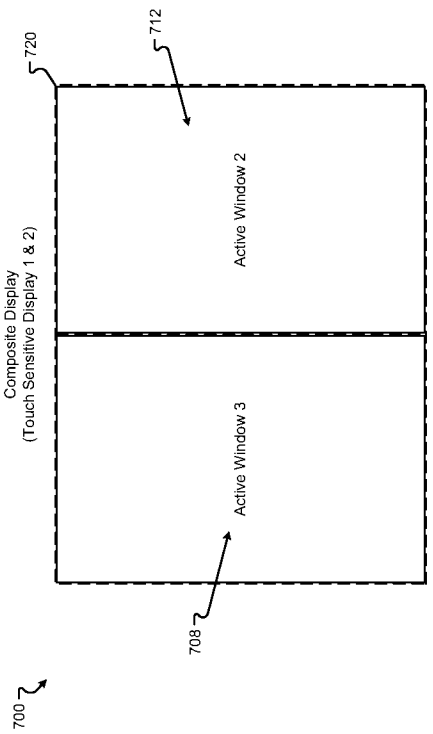


FIG. 7G

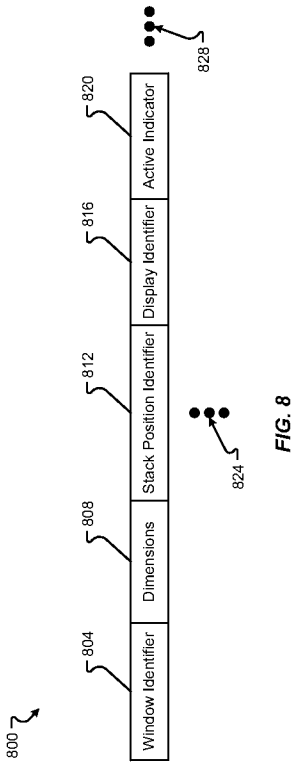


FIG. 8

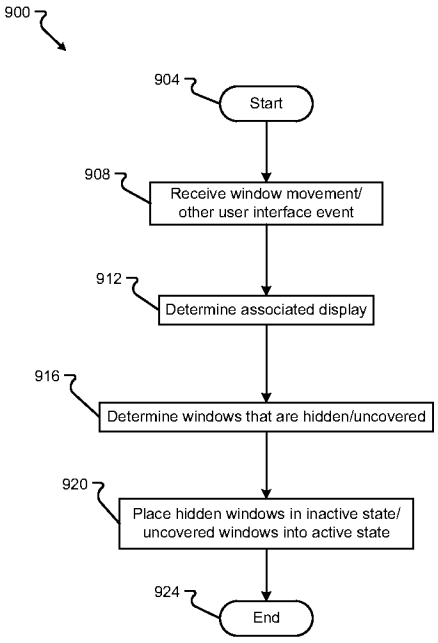


FIG. 9

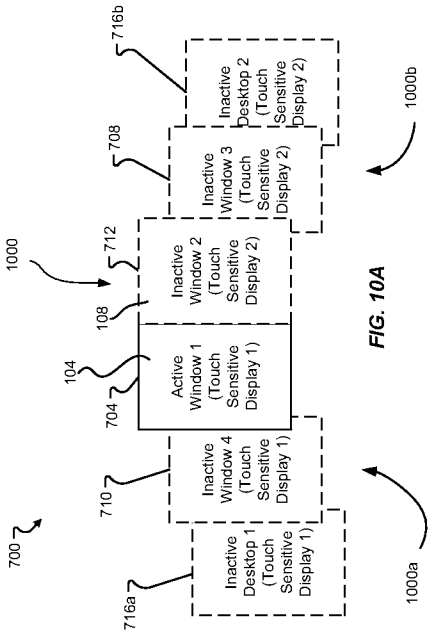


FIG. 10A

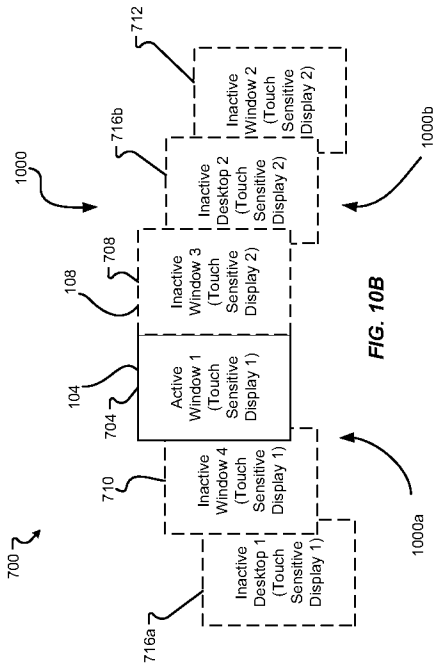


FIG. 10B

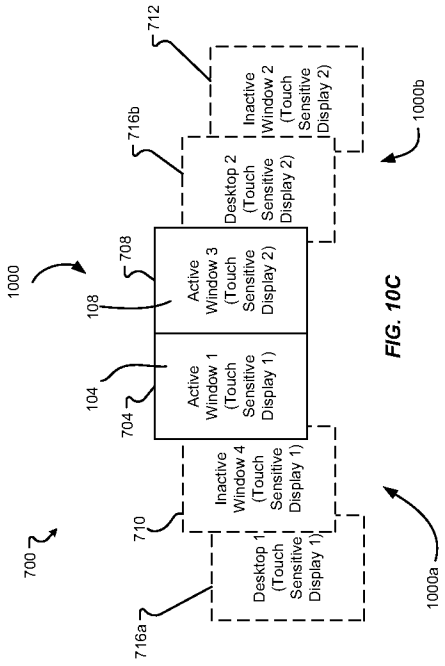


FIG. 10C

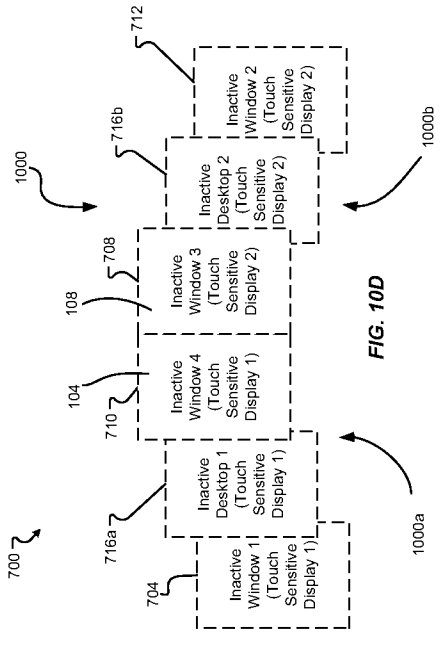


FIG. 10D

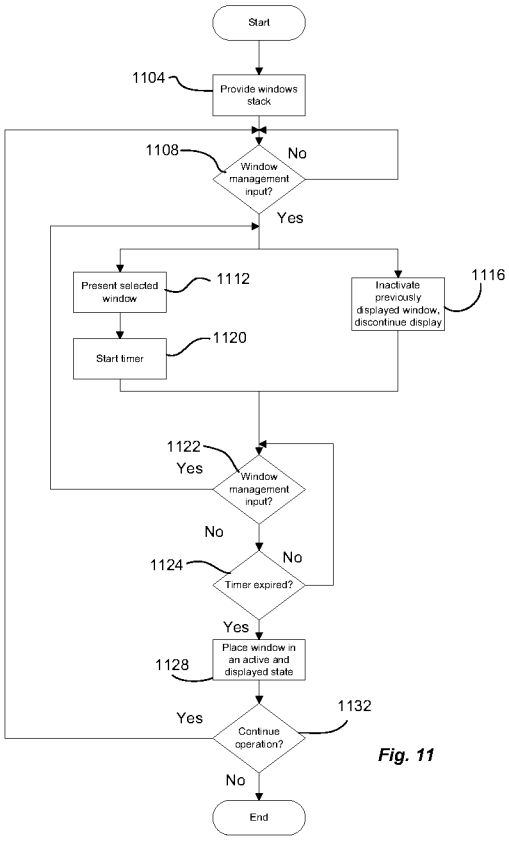


Fig. 11

METHOD AND APPARATUS FOR MOVING DISPLAY DURING A DEVICE FLIP

[0100] The device 100 manages desktops and/or windows within a user interface 700, as shown in Figs. 7A through 7G. A user interface 700 is a logical arrangement of active and inactive windows for a multi-screen device. For example, the user interface 700 may be logically similar to a deck of cards or a film strip, where one or more windows or desktops are arranged in order, as shown in Fig. 7A. An active window is a window that is currently being displayed on at least one of the touch sensitive displays 110, 114. For example, window 1 704 and window 3 708 are active windows and are displayed on touch sensitive displays 110 and 114, respectively. An inactive window is a window that was opened and displayed but is now "behind" an active window and not being displayed. In embodiments, an inactive window may be for an application that is suspended in a "sleep state", and thus, the window is not displaying active content. For example, window 2 712 is an inactive window in the sleep state.

[0101] A user interface 700 may have various arrangements or organizational structures. In the embodiment shown in Fig. 7A, the device 100 includes a set of windows/display objects or windows stack associated with a first touch sensitive display 110 and a second set of windows/display objects or windows stack associated with a second touch sensitive display 114. Thus, each touch sensitive display 110, 114 can have an associated set of windows/display objects. The two sets of windows/display objects may have different numbers of windows and may be arranged differently. Further, the two sets of windows/display objects can also be identified differently and managed separately. Thus, the first set of windows/display objects can be arranged in order from a first window to a last window. Likewise, the second set of windows/display objects can be arranged from a first window to a last window, but may have more or fewer windows. A logical data structure for managing the sets of windows/display objects may be as described in conjunction with Fig. 8.

[0102] When a window is moved, the newly moved window is repositioned within the user interface 700. Where and how the window is positioned within the user interface 700 can be a function of the orientation of the device 100, the context of what programs, functions, software, etc. are being executed on the device 100, how the user interface 700 is arranged when a new window is opened, when a window is moved, etc. To change the window position in the user interface 700, the window is moved by a user and information associated with the window can change. With this information, a logical data structure for the window can be changed and

stored. When user interface or other events or tasks change the arrangement of windows, the user interface 700 can be changed to reflect the change in arrangement. It should be noted that these same concepts described above can be used to manage the one or more desktops for the device 100.

[0103] For example, an arrangement of windows in a user interface can be shown in "profile" views in Figs. 7B through 7G. A top view is shown in Figs. 7B and 7G. Side views are shown in Figs. 7C-7F. Here, the logical arrangement of the user interface 700 is shown where, the active windows 704 and 708 are shown in Fig. 7B. The windows 704 and 708 can be the top most windows in portions of a composite display 720, which represents the total area of all the touch sensitive displays 110,114. As shown in Figs. 7C and 7D, window 1 704 and window 3 708 are at the top of the set of windows/display objects. The windows "beneath" the top windows, window 1 704 and window 3 708, are not active. Rather, window 2 712, which is not displayed, is inactive or in a sleep state. In a sleep state, the window 2 712 does not receive inputs. Further, the window need not generate a user interface window. As such, by setting the window 2 712 inactive, the processor saves on processor cycles and required code execution. When a window is "awakened", it is brought back to an active status, begins receiving inputs, and renders a window user interface. As shown in Figs. 7C and 7D, window 2 712 is below window 3 708 and not displayed, which changes in the embodiment shown in Figs. 7E and 7F.

[0104] In Figs. 7E through 7G, window 3 708 is moved from the second touch sensitive display 114 onto a first touch sensitive display 110 by gesture or other input to the device 110. In moving window 3 704, window 1 704 is covered and becomes inactive. Window 2 712 is exposed and becomes active. When a window 1 704 becomes inactive, a bitmap of the last view of the window is stored. The bitmap can be used when windows transfer onto the active display(s). Thus, the window appears to be active when transitioning but is not actually rendering a user interface until the transition is complete, and the window becomes active. Window 2 712 appears as the bitmap until window 3 708 has completed the move to the first touch sensitive display 110, and window 2 becomes active, as shown on Fig. 7G.

[0105] A logical data structure 800 for managing the arrangement of windows or desktops in a window user interface 700 is shown in Fig. 8. The logical data structure 800 can be any data structure used to store data whether an object, record, file, etc. The logical data structure 800 can be stored in any type of database or data storage system, regardless of protocol or standard. In

embodiments, the logical data structure 800 includes one or more portions, fields, attributes, etc. that store data in a logical arrangement that allows for easy storage and retrieval of the information. Hereinafrer, these one or more portions, fields, attributes, etc. shall be described simply as fields. The fields can store data for a window identifier 804, a dimension 808, a window stack position identifier 812, a display identifier 816, and/or an active indicator 820. Each window in a window user interface 700 can have an associated logical data structure 800. While only a single logical data structure 800 is shown in Fig. 8, there may be more or fewer logical data structures 800 used with a window user interface 700 (based on the number of windows or desktops in the user interface 700), as represented by ellipses 824. Further, there may be more or fewer fields than those shown in Fig. 8, as represented by ellipses 828.

[0106] A window identifier 804 can include any identifier (ID) that uniquely identifies the associated window in relation to other windows in the window user interface 700. The window identifier 804 can be a globally unique identifier (GUID), a numeric ID, an alphanumeric ID, or other type of identifier. In embodiments, the window identifier 804 can be one, two, or any number of digits based on the number of windows that can be opened. In alternative embodiments, the size of the window identifier 804 may change based on the number of windows opened. While the window is open, the window identifier 804 may be static and remain unchanged.

[0107] Dimensions 808 can include dimensions for the window. The dimensions 808 can limit the window to a portion or all of the composite display 720. Thus, the dimensions 808 can be a set of display coordinates for the window. For example, the window can include a first set of coordinates for a first corner and a second set of coordinates for a second corner. In other embodiments, the dimensions 808 can include one set of coordinates and a length and width, measured in pixels or other measure.

[0108] A window stack position identifier 812 can be any identifier that can identify the position in the window stack for the window. The window stack position identifier 812 can be a GUID, a numeric ID, an alphanumeric ID, or other type of identifier. Each window or desktop can include a window stack position identifier 812. For example, as shown in Fig. 7C, window 3 708, in a window stack, can have a window stack position identifier 812 of 1 identifying that window 708 is the first window in the window stack and the active window. Similarly, window 2 712 can have a window stack position identifier 812 of 2 representing that window 712 is the

second window in the window stack. Window 1 704 can also have a window stack position identifier 812 of 3 representing that window 704 is the third window in the window stack.

[0109] A display identifier 816 can identify that the window or desktop is associated with either the first display 110 or the second display 114. While this display identifier 816 may not be needed for a dual user interface 700 system, as shown in Fig. 7A, the display identifier 816 can indicate whether a window in the serial user interface 700 is displayed on a particular display. Thus, window 3 708 may have a display identifier 816 of 2 in Fig. 7D. Window 1 704 can have a display identifier 816 of 1 for the first display 110. However, in alternative embodiments, the window may have two display indicators 816 that represent that the window is displayed on both of the displays 110, 114 or may have a single identifier identifying the composite display 720.

[0110] Similar to the display identifier 816, an active indicator 820 may not be needed with the dual user interface 700 system. In the device 100, the active indicator 820 can indicate which window(s) in the user interface 700 is being displayed. Thus, window 1 704 may have an active indicator 820 in Fig. 7D but may not have an active indicator in Fig. 7E.

[0111] An embodiment of a method 900 for creating a window user interface 700 is shown in Fig. 9. While a general order for the steps of the method 900 is shown in Fig. 9. Generally, the method 900 starts with a start operation 904 and ends with an end operation 924. The method 900 can include more or fewer steps or can arrange the order of the steps differently than those shown in Fig. 9. The method 900 can be executed as a set of computer-executable instructions executed by a computer system and encoded or stored on a computer readable medium. Hereinafrer, the method 900 shall be explained with reference to the systems, components, modules, software, data structures, user interfaces, etc. described in conjunction with Figs. 1-8.

[0112] A multi-screen device 100 can receive a movement of a window or other user interface event (such as the closure of a window, opening of a new window, revealing a desktop, etc.), in step 908. While other user interface events are possible, the process 900 will be explained using a window movement as an example. One skilled in the art will understand how to apply the process 900 to other types of user interface events. In embodiments, the multi-screen device 100 can receive a window movement by receiving an input from the touch sensitive display 110 or 114, the configurable area 112 or 116, a gesture capture region 120 or 124, or some other hardware sensor operable to receive user interface inputs. The display controller 544 may send

the input signal to the processor 204 for processing. The processor 204 can interpret the input signal as a gesture, described in conjunction with Figs. 4A-4F, or other user input that moves an active or inactive window or moves a desktop. Upon receiving the gesture, the processor 204 can move the window. For example, window 3 708 can be moved onto the top of the first touch sensitive display 110, which makes window 2 712 active, as shown in Figs. 7E through 7G.

[0113] In embodiments, the processor 204 places the user interface interaction in the buffer 556 to be acted upon by the Input Management Module 536. Further, the processor 204 activates the inactive application, *i.e.*, activates the application ftom a suspended or sleep state. The application management class 540 may bind the application 564 to current display parameters. The device state module 574 of the Multi-Display Management class 524 may then determine how to rearrange the application in the user interface 700. Further, the device state module 574 may time the movement of a window. For example, before inactivating or activating a window. the device state module 574 may execute a timer. Only after the move is complete and the timer has expired will the device state module 574 activate or inactivate a window. The timer ensures that, if the user is flipping through multiple windows, then a window will not become active or other windows will not be put to sleep until the rapid movement of several windows ceases. the timer saves processor resources in not having to render user interfaces that will only be covered shortly thereafter.

[0114] The device state module 574 can first determine if the moved window has changed position in the user interface 700, in step 912. In other words, the device state module 574 determines to what position on the user interface, e.g., which display or to what position in a window stack, the window has been moved. Further, the device state module 574 may determine whether the movement of the window caused some other window to be covered in the user interface 700, in step 916. For example, the movement of window 3 708, show in Figs. 7E through 7G has changed the position of window 3 708 to the first touch sensitive display 110, has covered window 1 704, and has uncovered window 2 712. The device state module 574 determines these types of effects based on a window movement or other user interface event (e.g., opening a new window, closing a window, revealing the desktop, etc.).

[0115] Further, the display configuration module 568 places the hidden windows into a sleep state and activates uncovered windows, in step 920. Thus, the display configuration module 568 can change the active indicator 820 for the windows that are uncovered or covered in the user

interface 700. For example, window 1 704, which is covered in the embodiment shown in Fig. 7F, will have the active indicator eliminated to put the window to sleep. In the inactive or sleep state, window 1 stops receiving user interface inputs and stops rendering a user interface window. Further, window 2 712 has its active indicator set to acknowledge that window 2 712 is now active, receiving inputs, and rendering a window user interface. This sleep state information is passed to the Task Management Module 540 to instruct the Window Management Module 532 to render the new user interface 700 with the new active windows displayed and the other windows placed in the sleep state.

[0116] Fig. 10A depicts an output of the device 100 in an open state. In this example, first and second windows 704 and 712 are displayed by a first or primary screen 104 and a second or secondary screen 108 respectively. Moreover, in this example, the first window 704 is active, and the second window 712 is inactive.

[0117] In accordance with embodiments of the present invention, when the device 100 is placed in a closed state, operation of the secondary screen 108 is typically discontinued. However, in accordance with other embodiments of the present invention, a secondary closed mode can be activated, according to which the information that had been displayed in the primary screen 104 is displayed by the secondary screen 108. The secondary closed mode can be activated in various ways. For example, an application associated with a primary window 104 (e.g., window 1 704) can include instructions that cause the application window to be displayed by the secondary screen 108 when the device 100 is placed in the closed position or mode. Accordingly, this flip and slide action performed when entering the secondary closed mode can be initiated automatically when the device 100 is placed in the closed mode. In accordance with other embodiments of the present invention, the secondary closed mode can be activated manually. For example, a user can manually initiate the secondary closed mode by performing a twist and tap operation that includes closing the device, positioning the device so that the secondary screen 108 is in view of the user, and tapping the secondary screen. In accordance with a further embodiment, a user input comprising double tapping the gesture region or gesture capture area 120 of the primary screen 104 can provide a modal or sub-window that allows the user to enter an input to switch the active display from the primary screen 104 to the secondary screen 108. In accordance with other embodiments, activation of the secondary single screen

mode can be performed through a menu item, or through the selection of a user interface toggle or other object or input presenting an option to enter the secondary closed mode.

[0118] As shown in Fig. 10B, in accordance with at least some embodiments of the present invention, the primary screen 104 may display an instruction message when the device 100 is closed and the secondary closed mode is entered. For purposes of illustration, and without limitation, the instruction screen 1004 may display a text message such as "please turn over" to the user on the primary screen 104. Moreover, the instruction message 1004 may be discontinued after a predetermined period of time, as counted by a timer, has elapsed. After that predetermined period of time has elapsed, the primary screen 104 can be placed in an inactive or idle state.

[0119] With reference now to Fig. 10C, after entering the secondary closed mode, the window 704 that had been presented by the primary screen 104 while the device 100 was in the open mode or position (as shown in Fig. 10A) is presented by the secondary screen 108. In addition to presenting the active window on the secondary screen 108, the active window can be associated with the set of windows for the secondary screen 108. Accordingly, windowing management inputs entered while the device 100 is in the secondary closed mode operate with respect to the set of windows associated with the secondary screen 108.

[0120] As shown in Fig. 10D, in response to returning the device 100 to the open state following operation in the secondary closed mode, operation of the primary window is reinitiated or enabled. In accordance with embodiments of the present invention, when the device 100 is opened after being in the secondary closed mode, the application or window at the top of the set of windows associated with the primary screen 104 is displayed by the primary screen 104, while the active window continues to be displayed by the secondary screen 108. In addition, when the device 100 is returned to the open mode, the secondary closed mode is exited. More particularly, the device 100 will follow normal operation if it is again placed in the closed mode, with only the primary screen 104 active after the device 100 is placed in the closed state. Accordingly, absent a further instruction to activate the secondary closed mode provided as user input or in response to application programming associated with an active window as described above, the secondary closed mode will not be reinitiated. The secondary closed mode is therefore considered a temporary state of the device 100. Moreover, apart ftom the active window (e.g., the first window 704) being placed in the set of windows associated with the secondary screen

108, the effects of the secondary closed mode are terminated after the device 100 has been placed back into the open position.

[0121] Fig. 11 illustrates aspects of a method for moving the active display when the device 100 is placed in a closed mode or state in accordance with embodiments of the present invention. Initially, at step 1104, the device 100 is in an open state or mode, with an active window presented on the primary screen 104. The secondary screen 108 typically displays an inactive window or a desktop. Alternatively, the secondary screen 108 can be in an idle or powered off state. At step 1108, a determination is made as to whether user input in the form of placing the device 100 in a closed state or mode has been received. Until the device is closed, the process may idle at step 1108.

[0122] After the device has been placed in the closed state, a determination can be made as to whether an input or instruction to enter a secondary closed mode has been received (step 1112). As described elsewhere herein, input to enter the secondary closed mode can be in the form of a manual selection or input provided by a user. Alternatively, instructions to enter the secondary closed mode can be provided as part of the instructions associated with the active application. If such input has not been received, the device is operated in the normal closed screen mode, in which the primary screen 104 is active, and the secondary screen 108 is inactive (step 1114). The process can then end.

[0123] Where an input or instruction to enter the secondary closed mode has been received, the active window is presented on the secondary screen 108, and that window is associated with the set of windows for the secondary screen 108 (step 1116). In accordance with at least some embodiments of the present invention, an instruction message can be presented on the primary screen 104 (step 1120). This instruction generally includes a message to the user to turn or flip the device over so that the secondary screen 108 is in the view of the user, in case the user is viewing the primary screen 104 when the secondary closed mode is entered. Moreover, this instruction can be delivered as text presented by the primary screen 104. If presented, the instructions can be discontinued, and the primary screen can be placed in an idle or powered off mode, after a predetermined period of time has elapsed since the secondary closed mode was entered.

[0124] At step 1124, a determination can be made as to whether the device 100 has been placed in an open state. If the device 100 has not been opened, the device 100 continues to

operate in the secondary closed mode (step 1128). In the secondary closed mode, window management operations are performed with respect to the set of windows associated with the secondary screen 108.

[0125] If the device 100 has been opened, the secondary closed mode is discontinued (step 1132). After the secondary closed mode has been discontinued and the device 100 is in the open state, the active window continues to be presented by the secondary screen 108. In addition, the window presented by the secondary screen remains a part of the set of windows associated with the secondary screen 108, unless windowing management input moving that window to the set of windows associated with the primary screen is received. The window that was on the top of the stack or set of windows associated with the primary screen 104, after the active window was moved to the stack or set of windows associated with the secondary screen 108, is displayed by the primary window 104. After reentering the open mode, the process for operating in the secondary closed mode ends.

[0126] In accordance with embodiments of the present invention, after the secondary closed mode has been exited, closing the device does not reactivate the secondary closed mode, absent input or instructions to do so, as described above. Accordingly, the secondary closed mode is therefore a temporary device state.

[0127] In accordance with embodiments of the present invention, the method disclosed herein can be performed by the execution of application programming or other software components stored in memory 208, 508 by a processor 204, 504. For instance the operating system 516, a component or components of the framework 520, and/or a component or components of the multi-display management module 524 can include or provide functionality to operate the device 100 in a secondary single screen mode and associated modes as described herein. Moreover, embodiments of the present invention can provide the described method and functionality through the operation of a user interface 700, and an arrangement of windows including windows arranged in at least first and second sets, as described herein.

[0128] Although the method has, at least in some examples, been described in connection with a device having a primary 104 and secondary 108 screens, it should be appreciated by one of skill in the art after consideration of the present disclosure that embodiments of the present invention can be implemented in connection with more than two screens of a device 100. In

addition, embodiments of the present invention can be implemented in connection with a device 100 presenting or implementing multiple virtual screens in addition to multiple physical screens.

[0129] Various other exemplary systems and methods of this disclosure have been described in relation to the operation of a device in a closed state. However, such other systems and methods can be applied to operation of a single screen device, or alternatively to a device with more than two screens. In addition, embodiments of the present invention can be implemented in connection with a device 100 presenting or implementing multiple virtual screens.

[0130] Although embodiments of systems and methods have been described in connection with various structures and attributes, in order to avoid unnecessarily obscuring the present disclosure, the preceding description omits a number of known structures and devices. This omission is not to be construed as a limitation of the scopes of the claims. Specific details are set forth to provide an understanding of the present disclosure. It should however be appreciated that the present disclosure may be practiced in a variety of ways beyond the specific detail set forth herein.

[0131] Furthermore, while the exemplary aspects, embodiments, and/or configurations illustrated herein show the various components of the system collocated, certain components of the system can be located remotely, at distant portions of a distributed network, such as a LAN and/or the Internet, or within a dedicated system. Thus, it should be appreciated, that the components of the system can be combined in to one or more devices, such as the software modules of Figs. 5A and 5B, or collocated on a particular node of a distributed network, such as an analog and/or digital telecommunications network, a packet-switch network, or a circuit-switched network. It will be appreciated from the preceding description, and for reasons of computational efficiency, that the components of the system can be arranged at any location within a distributed network of components without affecting the operation of the system. For example, the various components can be located in a switch such as a PBX and media server, gateway, in one or more communications devices, at one or more users' premises, or some combination thereof. Similarly, one or more functional portions of the system could be distributed between a telecommunications device(s) and an associated computing device.

[0132] Furthermore, it should be appreciated that the various links connecting the elements can be wired or wireless links, or any combination thereof, or any other known or later developed element(s) that is capable of supplying and/or communicating data to and from the connected

elements. These wired or wireless links can also be secure links and may be capable of communicating encrypted information. Transmission media used as links, for example, can be any suitable carrier for electrical signals, including coaxial cables, copper wire and fiber optics, and may take the form of acoustic or light waves, such as those generated during radio-wave and infra-red data communications.

[0133] Also, while the flowcharts have been discussed and illustrated in relation to a particular sequence of events, it should be appreciated that changes, additions, and omissions to this sequence can occur without materially affecting the operation of the disclosed embodiments, configuration, and aspects.

[0134] A number of variations and modifications of the disclosure can be used. It would be possible to provide for some features of the disclosure without providing others. For example in one alternative embodiment, the window stack maintained in connection with the user interface 700 resembles a carousel or rolodex rather than a deck of cards. Thus, windows may circulate from one touch sensitive display 110 to another touch sensitive display 114. For example, window 6 724 may be pushed to the right and end up at the end of stack 764 behind window 716. If the stack continues to move to the right, eventually window 6 724 would appear on the second touch sensitive display 114 even though the window 724 was opened in the first touch sensitive display 110. These movements and changes in the stacks may be managed using the methods and logical data structures discussed above. In another alternative embodiment, there may be other arrangements for the window stack.

[0135] In yet another embodiment, the systems and methods of this disclosure can be implemented in conjunction with a special purpose computer, a programmed microprocessor or microcontroller and peripheral integrated circuit element(s), an ASIC or other integrated circuit, a digital signal processor, a hard-wired electronic or logic circuit such as discrete element circuit, a programmable logic device or gate array such as PLD, PLA, FPGA, PAL, special purpose computer, any comparable means, or the like. In general, any device(s) or means capable of implementing the methodology illustrated herein can be used to implement the various aspects of this disclosure. Exemplary hardware that can be used for the disclosed embodiments, configurations and aspects includes computers, handheld devices, telephones (e.g., cellular, Internet enabled, digital, analog, hybrids, and others), and other hardware known in the art. Some of these devices include processors (e.g., a single or multiple microprocessors), memory,

nonvolatile storage, input devices, and output devices. Furthermore, alternative software implementations including, but not limited to, distributed processing or component/object distributed processing, parallel processing, or virtual machine processing can also be constructed to implement the methods described herein.

[0136] In yet another embodiment, the disclosed methods may be readily implemented in conjunction with software using object or object-oriented software development environments that provide portable source code that can be used on a variety of computer or workstation platforms. Alternatively, the disclosed system may be implemented partially or fully in hardware using standard logic circuits or VLSI design. Whether software or hardware is used to implement the systems in accordance with this disclosure is dependent on the speed and/or efficiency requirements of the system, the particular function, and the particular software or hardware systems or microprocessor or microcomputer systems being utilized.

[0137] In yet another embodiment, the disclosed methods may be partially implemented in software that can be stored on a storage medium, executed on programmed general-purpose computer with the cooperation of a controller and memory, a special purpose computer, a microprocessor, or the like. In these instances, the systems and methods of this disclosure can be implemented as program embedded on personal computer such as an applet, JAVA® or CGI script, as a resource residing on a server or computer workstation, as a routine embedded in a dedicated measurement system, system component, or the like. The system can also be implemented by physically incorporating the system and/or method into a software and/or hardware system.

[0138] Although the present disclosure describes components and functions implemented in the aspects, embodiments, and/or configurations with reference to particular standards and protocols, the aspects, embodiments, and/or configurations are not limited to such standards and protocols. Other similar standards and protocols not mentioned herein are in existence and are considered to be included in the present disclosure. Moreover, the standards and protocols mentioned herein and other similar standards and protocols not mentioned herein are periodically superseded by faster or more effective equivalents having essentially the same functions. Such replacement standards and protocols having the same functions are considered equivalents included in the present disclosure.

[0139] The present disclosure, in various aspects, embodiments, and/or configurations, includes components, methods, processes, systems and/or apparatus substantially as depicted and described herein, including various aspects, embodiments, configurations embodiments, subcombinations, and/or subsets thereof. Those of skill in the art will understand how to make and use the disclosed aspects, embodiments, and/or configurations after understanding the present disclosure. The present disclosure, in various aspects, embodiments, and/or configurations, includes providing devices and processes in the absence of items not depicted and/or described herein or in various aspects, embodiments, and/or configurations hereof, including in the absence of such items as may have been used in previous devices or processes, e.g., for improving performance, achieving ease and\or reducing cost of implementation.

[0140] The foregoing discussion has been presented for purposes of illustration and description. The foregoing is not intended to limit the disclosure to the form or forms disclosed herein. In the foregoing Detailed Description for example, various features of the disclosure are grouped together in one or more aspects, embodiments, and/or configurations for the purpose of streamlining the disclosure. The features of the aspects, embodiments, and/or configurations of the disclosure may be combined in alternate aspects, embodiments, and/or configurations other than those discussed above. This method of disclosure is not to be interpreted as reflecting an intention that the claims require more features than are expressly recited in each claim. Rather, as the following claims reflect, inventive aspects lie in less than all features of a single foregoing disclosed aspect, embodiment, and/or configuration. Thus, the following claims are hereby incorporated into this Detailed Description, with each claim standing on its own as a separate preferred embodiment of the disclosure.

[0141] Moreover, though the description has included description of one or more aspects, embodiments, and/or configurations and certain variations and modifications, other variations, combinations, and modifications are within the scope of the disclosure, e.g., as may be within the skill and knowledge of those in the art, after understanding the present disclosure. It is intended to obtain rights which include alternative aspects, embodiments, and/or configurations to the extent permitted, including alternate, interchangeable and/or equivalent structures, functions, ranges or steps to those claimed, whether or not such alternate, interchangeable and/or equivalent structures, functions, ranges or steps are disclosed herein, and without intending to publicly dedicate any patentable subject matter.

What is claimed is:

1. A method for controlling a display of a window on a device, comprising:

presenting a first window by a primary screen of a multiple screen device, while the device is in an open mode;

receiving user input closing the device, thereby placing the device in a closed mode;

in response to the device being placed in the single screen mode, entering a secondary closed mode, wherein the first window is presented by a secondary screen of the multiple screen device.

2. The method of claim 1, further comprising:

in response to the device entering the secondary closed mode, discontinuing the presentation of the first window by the primary screen.

3. The method of claim 2, further comprising:

in response to the device entering the secondary closed mode, displaying an instruction on the primary screen.

4. The method of claim 3, wherein the displaying an instruction is discontinued after a predetermined period of time.

5. The method of claim 4, wherein the instruction is to turn the device over so that the secondary screen is in view of the user.

6. The method of claim 1, wherein entering the secondary closed mode in response to the device being placed in the closed mode is performed in accordance with application instructions associated with the first window.

7. The method of claim 1, wherein entering the secondary closed mode in response to the device being placed in the closed mode is performed in accordance with a user input directing that the first window be presented by the secondary screen.

8. The method of claim 1, wherein while presenting the first window by a primary screen the first window is associated with a first set of windows, and wherein in response to entering the secondary closed mode the first window is associated with a second set of windows.

9. The method of claim 1, further comprising:

after entering the secondary closed mode, receiving user input placing the device in the open mode, wherein the first window continues to be displayed by the secondary screen.

10. The method of claim 9, wherein the user input placing the device in the open mode after entering the secondary screen mode discontinues the secondary closed mode.

11. A device, comprising:

at least first and second screens;

a hinge interconnecting the first and second screens;

memory;

a processor in communication with the at least first and second screens and the memory;

application programming stored in the memory and executed by the processor, wherein the application programming is operable to:

maintain at least a first and second sets of windows, wherein the first and second sets of windows are stored in the memory, wherein the first set of windows is associated with at least a first window and is associated with the first screen, and wherein the second set of windows is associated with at least a second window and is associated with the second screen;

present the first window in the first screen while the device is in an initial, open state;

receive a first user input, wherein the first user input includes the user placing the device in a closed state;

in response to the device being placed in the closed state, enter a secondary closed mode, wherein the first window is presented by a secondary screen of the device.

12. The device of claim 11, wherein the application programming is further operable to display an instruction on the primary screen in response to the device entering the secondary closed mode.

13. The device of claim 11, wherein the application programming is further operable to disassociate the first window from the first set of windows and to associate the first window with the second set of windows.

14. The device of claim 11, further comprisinig:

a position sensor, wherein the application programming is further operable to detect the mode of the device in response to a signal from the position sensor.

15. The device of claim 11, further comprising:

at least one of a touch sensitive display area or a gesture capture area included in at least the second screen, wherein the application programming is further operable to display a second window included in the second set of windows in response to a window management operation received through one of the touch sensitive display area or the gesture capture area included in the second screen that dismisses the first window.

16. A computer readable medium having stored thereon computer-executable instructions, the computer executable instructions causing a processor to execute a method for controlling a user interface of a device, the computer-executable instructions comprising:

instructions to present a first window by a primary screen of the device, while the device is in an open mode;

instructions to detect receive user input closing the device, thereby placing the device in a closed mode;

in response to the device being placed in the closed mode, instructions to enter a secondary closed mode, wherein the first window is presented by a secondary screen of the device.

17. The computer readable medium of claim 16, further comprising:

instructions to maintain a first set of windows, wherein the first window is maintained in the first set of windows while the device is in the open mode;

instructions to maintain a second set of windows, wherein the first window is maintained in the second set of windows and not in the first set of windows while the device is in the secondary closed mode.

18. The computer readable medium of claim 17, the computer executable instructions further comprising:

instructions to present a second window in the secondary screen of the device in response to a windowing management input instructing that the presentation of the first window by the secondary screen be discontinued, wherein the second window is maintained in the second set of windows.

19. The computer readable medium of claim 18, the computer executable instructions further comprising:

instructions to continue to display the first window in the secondary screen of the device in response to user input returning the device to the open mode.

20. The computer readable medium of claim 19, wherein after the device is returned to the open mode the first window continues to be maintained in the second set of windows.

ABSTRACT

Systems and methods for moving a display during a device flip are provided. More particularly, in response to input or instructions selecting a secondary closed mode, operation of the device can be altered from a normal operating mode. In the secondary closed mode, when a device is moved from an open mode or state to a closed mode or state, the active window while the device was in the open mode is presented by a secondary screen, rather than a primary screen. The secondary closed mode is exited when the device is again placed in an open state or mode.

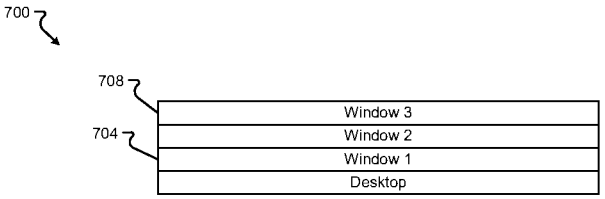


FIG. 7C

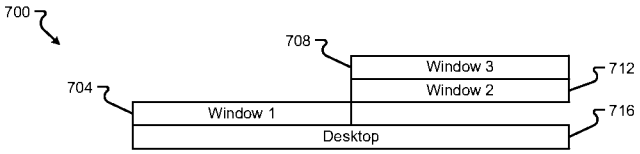


FIG. 7D

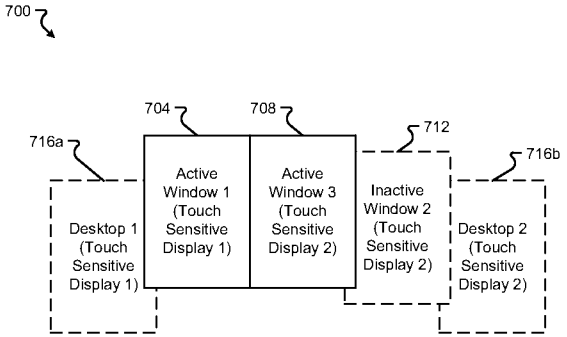


FIG. 7A

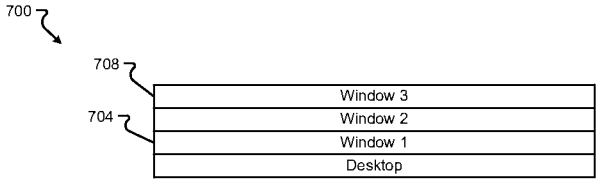


FIG. 7E

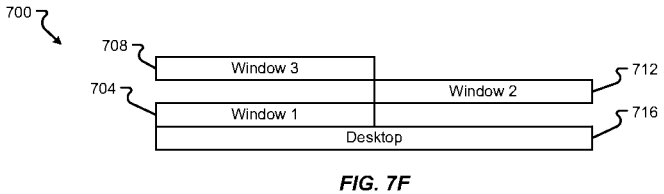


FIG. 7F

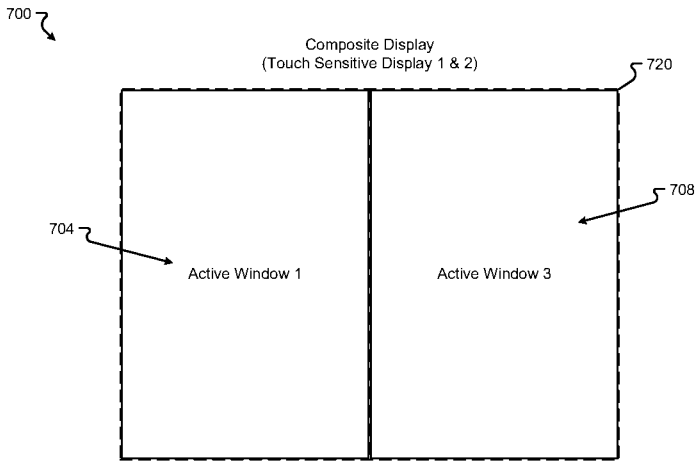


FIG. 7B

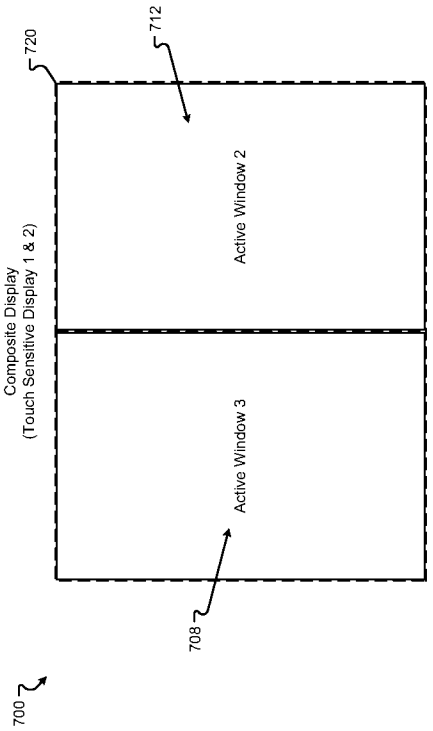


FIG. 7G

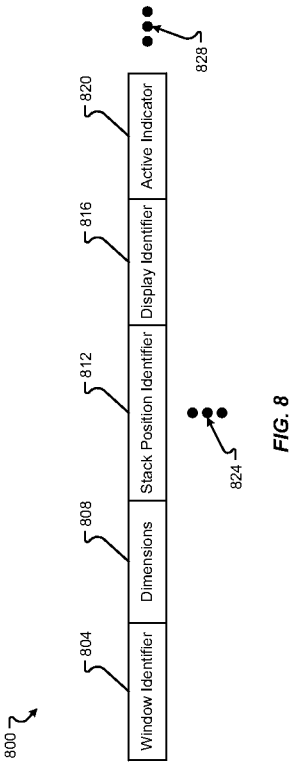


FIG. 8

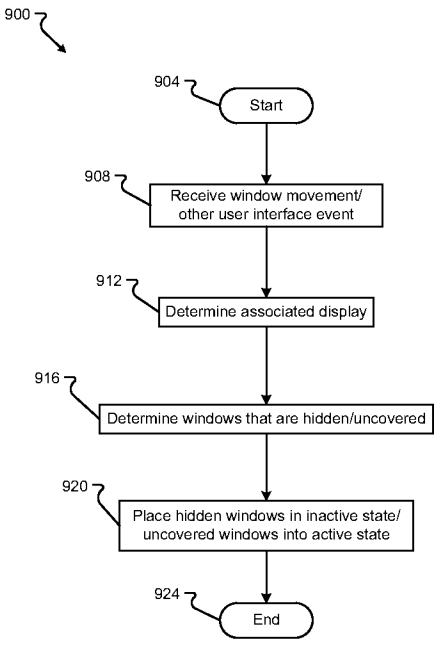


FIG. 9

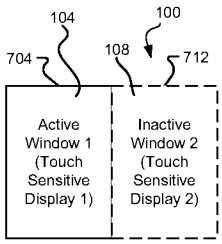


FIG. 10A

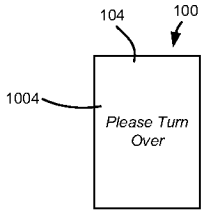


FIG. 10B

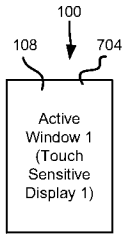


FIG. 10C

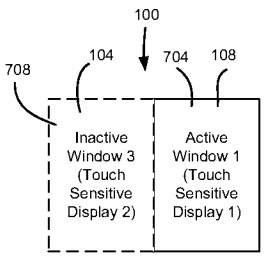


FIG. 10D

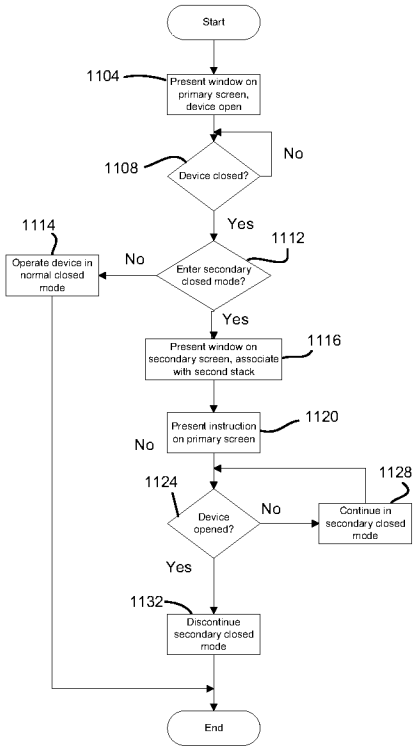

Start
1104
Present window on primary screen, device open
1108
Device closed?
No
Yes
1114
Operate device in normal closed mode
No
1112
Enter secondary closed mode?
Yes
1116
Present window on secondary screen, associate with second stack
1120
Present instruction on primary screen
No
1128
1124
Device opened?
No
Continue in secondary closed mode
Yes
1132
Discontinue secondary closed mode
End

Fig. 11

[0100] FIGS. 7A-7C illustrate an embodiment showing device 100 and how device 100 responds when a phone call is being received. FIG. 7A illustrates device 100 (with screens 104 and 108). Although FIGS. 7A-7C appear to have screens 10 Ford one way to the open position, this is being done merely for illustrative purposes. FIGS. 7A-7C are intended to illustrate operation of the device 100 when it is in a closed configuration as illustrated above in FIGS. 1H, 6A, 6B, 6D, or 6E. As noted in FIG. 7A screen 104 is in the ftont of the device while screen 108 is on the back of the device. In the embodiments described below, screen 104 is considered the primary display, while screen 108 is considered a secondary screen. It is noted that in other embodiments, the primary screen may be screen 108 and the secondary screen is screen 104.

[0101] FIG. 7A illustrates device 100 when an application has been launched and a window 704 of the application is displayed on the display 114 of screen 108. In FIG. 7A, device 100 is operating in secondary single screen mode, meaning that only one screen is active, namely the secondary screen 108. There are no windows displayed on display 110 of screen 104, and at least a portion of screen 104 is in active.

[0102] As shown in FIG. 7B, a phone call was received by device 100. As a result, a message 708 is displayed on display 114 of screen 108 indicating that the call has been received. The message 708, may in some embodiments in addition to indicating that the call is received, also indicate that the device 100 should be repositioned. For example, the message 708 may indicate that the device should be turned to view the display 110 on the primary screen 104.

[0103] As shown in FIG. 7C, a user interface 712 is displayed on display 114 of screen 104. The user interface 712 allows a user to handle the incoming call, such as by accepting the call or cancelling the call. As can be appreciated, a user may accept the call or cancel the call. If the call is accepted, the call will be ended after a conversation. Regardless of whether the call is accepted or canceled, device 100 will stop operating in the secondary single screen mode while the call is being received and handled. Instead, device 100 will operate in the primary single screen mode where the user interacts with screen 104, at least a portion of screen 108 is an active.

[0104] As also shown in FIG. 7C, the window 704 remains on display 114. A user can, in some embodiments, return to interacting with window 704 after the call ended. In other words, device 100 can be configured to automatically switch back to the secondary single screen mode after the call has ended.

[0105] FIGS. 8A-8C illustrates another embodiment of device 100 and its output in response to receiving a phone call. FIG. 8A illustrates device 100 (with screens 104 and 108). FIG. 8A illustrates device 100 when the application has been launched and the window 704 of the application is displayed on the display 114 of screen 108, similar to FIG. 7A. FIG. 8A illustrates device 100 is operated in the secondary single screen mode. There are no windows displayed on display 110 of screen 104. FIG. 8B illustrates that a phone call was received by device 100. As a result, message 708 is displayed on display 114 of screen 108 indicating that the call has been received. The message 708 also indicates that the device 100 should be repositioned.

[0106] As shown in FIG. 8C, user interface 712 is displayed on display 114 of screen 104. The user interface 712 allows a user to handle the incoming call, such as by accepting the call or cancelling the call. As also shown in FIG. 7C, the window 704 is no longer displayed on display 114 of screen 108. In this embodiment, device 100 is no longer operating in the secondary single screen mode, rather the device 100 is operating in primary single screen mode. For this reason, window 704 has been minimized and the minimized window 804 is displayed on display 110 of screen 104. After a user has finished with the phone call, the user can return to interacting with window 704 by selecting the minimized window 804 after the call ended. In this embodiment, device 100 does not automatically return to the secondary single screen mode. The user may enter the necessary input (e.g. selection of a menu(s) etc.) to return to the secondary single screen mode, but by default device 100 will remain in primary single screen mode after a call is ended.

[0107] FIGS. 9A-8C illustrates another embodiment of device 100 and its output in response to receiving a phone call. FIG. 9A illustrates device 100 (with screens 104 and 108). FIG. 9A illustrates device 100 when an application has been launched and a window 904 of the application is displayed on the display 110 of screen 104. FIG. 9A illustrates device 100 is operated in the primary single screen mode. There are no windows displayed on display 114 of screen 108. FIG. 9B illustrates that a phone call was received by device 100. As a result, user interface 712 is displayed on display 114 of screen 104.

[0108] As previously indicated, the user interface 712 allows a user to handle the incoming call, such as by accepting the call or cancelling the call. In FIG. 9B, a user has selected to cancel the incoming phone call. In other embodiments, a user can answer the phone call and engage in a conversation. The call is then ended after the conversation.

[0109] FIG. 7C illustrates that after the call is cancelled, or ended in embodiments, the window 904 is redisplayed on display 110 of screen 104. In this embodiment, device 100 continualy operates in the primary single screen mode. For this reason, window 904 is displayed on display 110 of screen 104 after the call was canceled or ended to allow the user to continue to interact with the open application. In this embodiment, device 100 does is not operated in secondary single screen mode. The user may enter the necessary input (e.g. selection of a menu(s) etc.) to begin operating in the secondary single screen mode, but because device 100 is operating in the primary single screen mode prior to the phone call, after the phone call, device 100 remains operating in primary single screen mode.

[0110] Referring now to FIG. 10, flow diagram 1000 in accordance with at least some embodiments of the present disclosure is shown and will be described. Flow 1000 is in embodiments performed by a device such as device 100. More specifically, one or more hardware or software components may be involved in performing flow 1000. In one embodiment, modules in middleware 520 (FIG. 5A) such as multi-display management class 524 (FIG. 5B) perform one or more of the steps of flows 1200 and 1300. In other embodiments, in addition to, or in lieu of, middleware 520 performing steps of flows 1200 and 1300, operating system kernel(s) 516a, processor 504, and/or drivers 512a-512b may also perform steps of flow 1000.

[0111] Flow 1000 is initiated at 1004. Flow 1000 passes ftom 1004 to optional step 1008 where input regarding call receiving behavior is received. In this step, a user may provide configuration information that indicates how a device should handle incoming calls and windows displayed on one or more displays when there is an incoming call. In embodiments, the device is device 100 (FIGS. 1-3), which includes multiple screens. The user may input any information that affects how a device 100 may handle incoming calls, including but not limited to, how applications are minimized, changes (if any) that occur to displays 104 and 108 when a phone call is received, and other information. In one embodiment, the input indicates that all calls will be handled using a user interface displayed on the primary screen.

[0112] From step 1008, flow 1000 passes to step 1012 where device 100 is operated in primary single screen mode. In this mode, a user interacts with a display on the primary screen. A user can for example launch applications. At step 1014, a window of an application is displayed on a

display of the primary screen, and a user can enter input, and utilize the application using the display window.

[0113] A first call is received at step 1016. In response, a user interface is displayed at step 1020 on the display of the primary screen so that the user can handle the call. As indicated above with respect to FIGS. 7B, 7C, and 7D, the user interface can include images that allow a user to answer call/cancel a call. A user can answer the call or cancel the call. At some point in time, however, the call is ended at step 1024. The call may be ended by the user selecting to cancel the call using the UI displayed at step 1020, or the call may be ended after a conversation.

[0114] After the conversation has ended at step 1024, flow passes to step 1026, where the first window that was previously displayed, before the call was received at step 1016, is redisplayed on a display of the primary screen. The device continues to operate in primary single screen mode, before the call, during the call, and after the call is ended. In primary single screen mode, at least a portion of a secondary screen(s) is inactive, meaning that input is not received ftom that portion of the secondary screen.

[0115] Flow 1000 passes to step 1028 where primary single screen mode is exited. Step 1028 is followed by step 1032 where the device begins to operate secondary single screen mode. In secondary single screen mode, at least a portion of the primary screen is inactive, meaning that input is not received ftom that portion of the primary screen. In the secondary single screen mode, a second application can be launched ftom the secondary screen. A second window would then be displayed on the secondary screen in response to launching the second application. At step 1034, a second window is displayed on the secondary screen as a result of having launched the second application.

[0116] At step 1036, a second call is received. Because the device is in secondary single screen mode, at step 1040 a message is displayed on the secondary screen indicating that the second call is received. In some embodiments, flow 1000 includes an optional step 1044 where in addition to displaying the message, indicating that the second call has been received, the message may also include information indicating that the device should be repositioned. For example, since the device is being used in the secondary single screen mode, the user may not be viewing the primary screen, where the phone call will actually be handled. At step 1044, the message may indicate to the user to rotate, turn, or otherwise reposition the device to allow the user to view the primary screen.

[0117] Following step 1044, at step 1048, a UI is displayed on a display of the primary screen in order to handle the second call. As indicated above, the UI may include features that allow the user to answer or cancel the call. Input for handling the call is received at 1052. Because the call is handled using the UI displayed on the primary screen, the secondary screen mode is exited at step 1056 and the devices is operated in primary single screen mode at step 1058.

[0118] At some point in time, however, the second call is ended at step 1060. The second call may be ended by the user selecting to cancel the call using the UI displayed at step 1048, or the call may be ended after a conversation.

[0119] Because, after the call, the device is still operating in secondary single screen mode, the second window that was previously displayed on the secondary screen, is now displayed on the primary screen at step 1064. This allows the user to continue to interact with the second application using the second window after the call is ended. Flow 1000 then ends at 1068.

[0120] It is noted that in other embodiments, the device can be configured to switch back to secondary single screen mode after the call is ended. In these embodiments, step 1064 would involve displaying the second window on the secondary screen. The user can then interact with the window on the secondary screen.

[0121] The exemplary systems and methods of this disclosure have been described in relation to FIGS 1-10. However, to avoid unnecessarily obscuring the present disclosure, the preceding description omits a number of known structures and devices. This omission is not to be construed as a limitation of the scopes of the claims. Specific details are set forth to provide an understanding of the present disclosure. It should however be appreciated that the present disclosure may be practiced in a variety of ways beyond the specific detail set forth herein.

[0122] Furthermore, while the exemplary aspects, embodiments, and/or configurations illustrated herein show the various components of the system collocated, certain components of the system can be located remotely, at distant portions of a distributed network, such as a LAN and/or the Internet, or within a dedicated system. Thus, it should be appreciated, that the components of the system can be combined in to one or more devices, such as a phone, computer, PDA, electronic book reader, gaming device, or collocated on a particular node of a distributed network, such as an analog and/or digital telecommunications network, a packet-switch network, or a circuit-switched network. It will be appreciated ftom the preceding description, and for reasons of computational efficiency, that the components of the system can

be arranged at any location within a distributed network of components without affecting the operation of the system. For example, the various components can be located in a switch such as a PBX and media server, gateway, in one or more communications devices, at one or more users' premises, or some combination thereof. Similarly, one or more functional portions of the system could be distributed between a telecommunications device(s) and an associated computing device.

[0123] Furthermore, it should be appreciated that the various links connecting the elements can be wired or wireless links, or any combination thereof, or any other known or later developed element(s) that is capable of supplying and/or communicating data to and ftom the connected elements. These wired or wireless links can also be secure links and may be capable of communicating encrypted information. Transmission media used as links, for example, can be any suitable carrier for electrical signals, including coaxial cables, copper wire and fiber optics, and may take the form of acoustic or light waves, such as those generated during radio-wave and infra-red data communications.

[0124] Also, while the flowcharts have been discussed and illustrated in relation to a particular sequence of events, it should be appreciated that changes, additions, and omissions to this sequence can occur without materially affecting the operation of the disclosed embodiments, configuration, and aspects.

[0125] A number of variations and modifications of the disclosure can be used. It would be possible to provide for some features of the disclosure without providing others.

[0126] For example in one alternative embodiment, all of the phone call receiving behavior information and information regarding launching applications may be preprogrammed into the device 100. In these embodiments, there would be no input received ftom a user regarding how a device 100 handles gestures indicating the launching of applications or a change in focus.

[0127] In other alternative embodiments, device 100 may be in any configuration described above with respect to FIGS. 6A-6J. In these embodiments, windows of open applications may be minimized, displayed, or modes changed in any of the described configurations.

[0128] In yet another embodiment, the systems and methods of this disclosure can be implemented in conjunction with a special purpose computer, a programmed microprocessor or microcontroller and peripheral integrated circuit element(s), an ASIC or other integrated circuit, a digital signal processor, a hard-wired electronic or logic circuit such as discrete element circuit,

a programmable logic device or gate array such as PLD, PLA, FPGA, PAL, special purpose computer, any comparable means, or the like. In general, any device(s) or means capable of implementing the methodology illustrated herein can be used to implement the various aspects of this disclosure. Exemplary hardware that can be used for the disclosed embodiments, configurations and aspects includes computers, handheld devices, telephones (e.g., cellular, Internet enabled, digital, analog, hybrids, and others), and other hardware known in the art. Some of these devices include processors (e.g., a single or multiple microprocessors), memory, nonvolatile storage, input devices, and output devices. Furthermore, alternative software implementations including, but not limited to, distributed processing or component/object distributed processing, parallel processing, or virtual machine processing can also be constructed to implement the methods described herein.

[0129] In yet another embodiment, the disclosed methods may be readily implemented in conjunction with software using object or object-oriented software development environments that provide portable source code that can be used on a variety of computer or workstation platforms. Alternatively, the disclosed system may be implemented partially or fully in hardware using standard logic circuits or VLSI design. Whether software or hardware is used to implement the systems in accordance with this disclosure is dependent on the speed and/or efficiency requirements of the system, the particular function, and the particular software or hardware systems or microprocessor or microcomputer systems being utilized.

[0130] In yet another embodiment, the disclosed methods may be partially implemented in software that can be stored on a storage medium, executed on programmed general-purpose computer with the cooperation of a controller and memory, a special purpose computer, a microprocessor, or the like. In these instances, the systems and methods of this disclosure can be implemented as program embedded on personal computer such as an applet, JAVA® or CGI script, as a resource residing on a server or computer workstation, as a routine embedded in a dedicated measurement system, system component, or the like. The system can also be implemented by physically incorporating the system and/or method into a software and/or hardware system.

[0131] Although the present disclosure describes components and functions implemented in the aspects, embodiments, and/or configurations with reference to particular standards and protocols, the aspects, embodiments, and/or configurations are not limited to such standards and

protocols. Other similar standards and protocols not mentioned herein are in existence and are considered to be included in the present disclosure. Moreover, the standards and protocols mentioned herein and other similar standards and protocols not mentioned herein are periodically superseded by faster or more effective equivalents having essentially the same functions. Such replacement standards and protocols having the same functions are considered equivalents included in the present disclosure.

[0132] The present disclosure, in various aspects, embodiments, and/or configurations, includes components, methods, processes, systems and/or apparatus substantially as depicted and described herein, including various aspects, embodiments, configurations embodiments, subcombinations, and/or subsets thereof. Those of skill in the art will understand how to make and use the disclosed aspects, embodiments, and/or configurations after understanding the present disclosure. The present disclosure, in various aspects, embodiments, and/or configurations, includes providing devices and processes in the absence of items not depicted and/or described herein or in various aspects, embodiments, and/or configurations hereof, including in the absence of such items as may have been used in previous devices or processes, e.g., for improving performance, achieving ease and\or reducing cost of implementation.

[0133] The foregoing discussion has been presented for purposes of illustration and description. The foregoing is not intended to limit the disclosure to the form or forms disclosed herein. In the foregoing Detailed Description for example, various features of the disclosure are grouped together in one or more aspects, embodiments, and/or configurations for the purpose of streamlining the disclosure. The features of the aspects, embodiments, and/or configurations of the disclosure may be combined in alternate aspects, embodiments, and/or configurations other than those discussed above. This method of disclosure is not to be interpreted as reflecting an intention that the claims require more features than are expressly recited in each claim. Rather, as the following claims reflect, inventive aspects lie in less than all features of a single foregoing disclosed aspect, embodiment, and/or configuration. Thus, the following claims are hereby incorporated into this Detailed Description, with each claim standing on its own as a separate preferred embodiment of the disclosure.

[0134] Moreover, though the description has included description of one or more aspects, embodiments, and/or configurations and certain variations and modifications, other variations, combinations, and modifications are within the scope of the disclosure, e.g., as may be within the

skill and knowledge of those in the art, after understanding the present disclosure. It is intended to obtain rights which include alternative aspects, embodiments, and/or configurations to the extent permitted, including alternate, interchangeable and/or equivalent structures, functions, ranges or steps to those claimed, whether or not such alternate, interchangeable and/or equivalent structures, functions, ranges or steps are disclosed herein, and without intending to publicly dedicate any patentable subject matter.

What is claimed is:

1. A method, comprising:
displaying a window of an application on a secondary screen of a multi-screen device;
receiving a call on the multi-screen device;
displaying on the secondary screen of the multi-screen device an indication that the call has been received; and
displaying, on the primary screen, a user interface that allows the user to answer the call.

2. The method of claim 1, wherein the indication includes a message indicating to a user that the primary screen should be viewed.

3. The method of claim 1, further comprising:
displaying a portion of the window on the primary display.

4. The method of claim 1, wherein the portion of the window comprises a minimized window.

5. The method of claim 1, further comprising:
receiving input from the primary screen.

6. The method of claim 5, wherein the input from the primary screen results in answering the call.

7. The method of claim 5, wherein the input from the primary screen results in ending the call.

8. The method of claim 7, in response to the input from the primary screen, displaying the window on a display of the primary screen.

9. The method of claim 8, further comprising:
in response to the input from the primary screen, changing the secondary screen to indicate that input is being received from the primary screen.

10. The method of claim 9, wherein the changing the secondary screen comprises darkening at least a portion of the secondary screen.

11. A non-transitory computer readable medium storing computer executable instructions that when executed by at least one processor perform a method comprising:
operating a multi-screen device in a secondary single screen mode, wherein in the secondary single screen mode, a window of an application is displayed on a secondary screen of a multi-screen device and a primary screen of the single screen device is darkened;
receiving a call on the multi-screen device;
displaying on the secondary screen of the multi-screen device an indication that the call has been received;
displaying, on the primary screen, a user interface that allows the user to answer the call;
receiving input by the primary screen; and
exiting the secondary single screen mode.

12. The non-transitory computer readable medium of claim 11, wherein input received by the primary screen results in answering the call.

13. The non-transitory computer readable medium of claim 11, wherein input received by the primary screen results in the call being canceled.

14. The non-transitory computer readable medium of claim 11, further comprising
afrer exiting the secondary single screen mode, operating the multi-screen device in a primary single screen mode.

15. The non-transitory computer readable medium of claim 14, further comprising
afrer exiting the secondary single screen mode, displaying the window of the application on the primary screen.

16. The non-transitory computer readable medium of claim 11, wherein the method further comprises:
afrer receiving the call, displaying a message indicating that the multi-screen device should be repositioned.

17. A dual screen communication device, comprising:
a first touch sensitive display of a first screen;
a second touch sensitive display of a second screen;
a computer readable medium that stores computer executable instructions that when executed by at least one processor perform a method comprising:
operating a multi-screen device in a primary single screen mode, wherein in the primary single screen mode, a first window of a first application is displayed on a primary screen of a multi-screen device and at least a portion of a secondary screen of the single screen device is darkened;
receiving a first call on the multi-screen device;
displaying, on the primary screen, a user interface that allows the user to answer the first call;
receiving first input by the primary screen to end the call;
exiting the primary single screen mode;
operating the multi-screen device in a secondary single screen mode, wherein in the secondary single screen mode, a second window of a second application is displayed on the secondary screen and at least a portion of the primary screen is darkened;
receiving a second call on the multi-screen device;
in response to receiving the second call:
displaying on the secondary screen of the multi-screen device an indication that the second call has been received;

displaying, on the primary screen, the user interface that allows the user to answer the second call;

receiving a second input by the primary screen; and

in response to receiving the second input by the primary screen, exiting the secondary single screen mode.

18. The device of claim 17, further comprising afrer exiting the secondary single screen mode, operating the multi-screen device in the primary single screen mode.

19. The device of claim 17, further comprising afrer exiting the secondary single screen mode, displaying the second window of the second application on the primary screen.

20. The device of claim 17, further comprising in response to receiving the second call, displaying a message indicating that the multi-screen device should be repositioned.

ABSTRACT

Embodiments are described for handling receipt of call in a multi-screen device. In embodiments, the device may be in a closed mode in which a primary screen is being used. A message regarding the incoming call is displayed on the primary screen to a user so that a user can decide whether to answer the call from the primary screen. If the device is being used in a closed secondary screen mode (with the user interacting with the secondary screen) when the call is received, a notice will be displayed to the user to turn the phone around to the primary screen so that the user can decide whether to answer the phone.

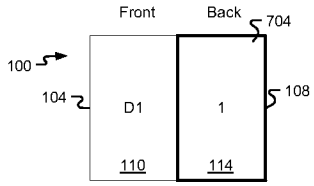


FIG. 7A

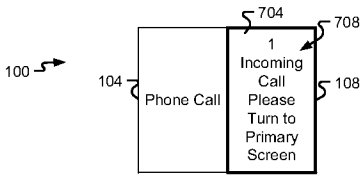


FIG. 7B

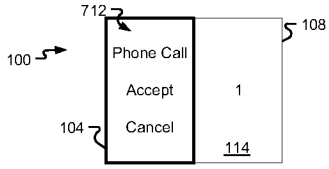


FIG. 7C

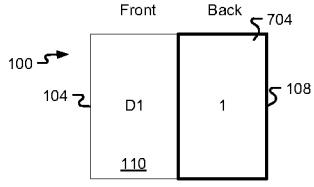


FIG. 8A

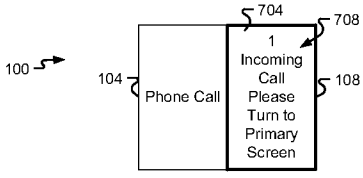


FIG. 8B

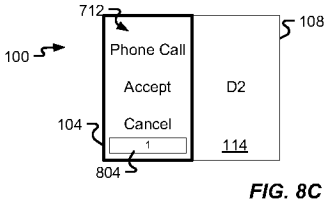


FIG. 8C

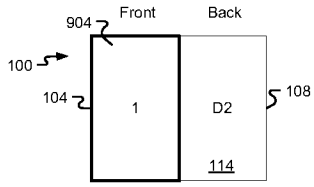


FIG. 9A

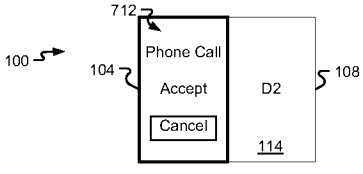


FIG. 9B

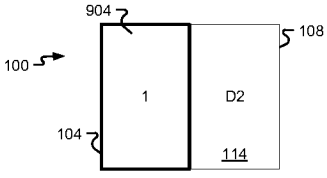
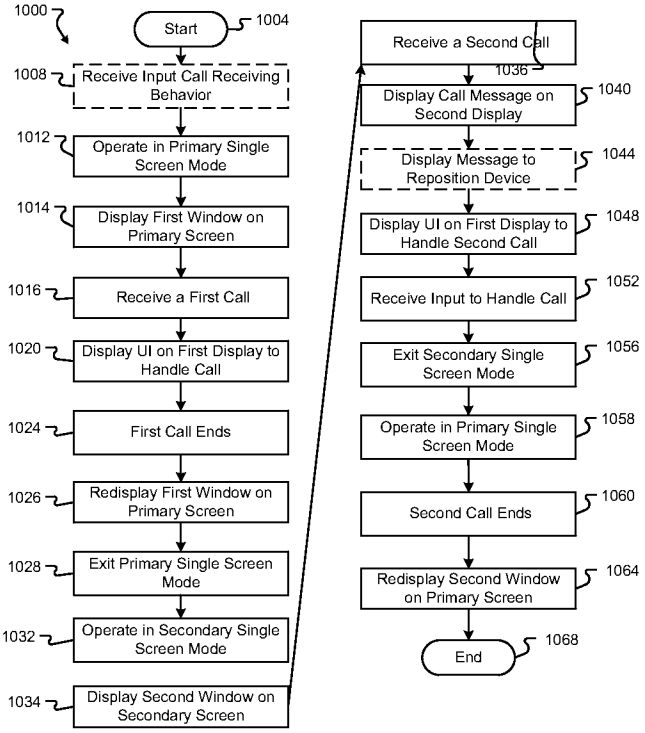


FIG. 9C



[0100] FIGS. 7A-7C illustrate an embodiment showing device 100 and how device 100 responds when applications are launched. FIG. 7A illustrates device 100 (with screens 104 and 108) when no applications have been launched and a desktop is displayed on both display 110 and display 114. As shown in FIG. 7B, a user can make a gesture 708, such as a tap, drag, or flick gesture on display 110 to indicate a request to launch an application. In some embodiments, the gesture may be made in gesture capture region 120 (e.g., gesture 708A). Device 100 receives the gesture 708 or 708A. In response to receiving the gesture, device 100 will launch the application. As shown in FIG 7C, the launching of the application results in an image 708 being displayed on display 110. The image 708 is for example an open window that provides a user interface. The window will have a view of the application and allow the user to interact with the launched application. As shown in FIG. 7C, the image 708 is displayed on display 110 because he input, i.e. gesture 704 was received on display 110.

[0101] It is noted that gesture 704 may be made on one or more portions of display 110, a configurable area of screen 104, or as shown in FIG 7B (gesture 708A) a gesture capture region 120. The gesture 708 may involve touching portions of screen 104 and or can involve only movement above the surface of portions of screen 104. In some embodiments, for the gesture to be recognized, contact is required to be made with the display 110 for a predetermined period of time.

[0102] In some embodiments, device 100 is configured to interpret different gestures as indications of different operations. For example, the device 100 may be configured to move an image when the gesture 708 is a drag gesture or flick gesture, but launch applications if gesture 708 is a flick, tap gesture, or double tap gesture. In embodiments, when an application is launched, any image displayed in response to the launching of the application (e.g., open window) is placed in focus. This allows a user to immediately begin to interact with the open application to the displayed image. In some embodiments, the location of the gesture 708, whether in gesture capture region 120 or some other portion of display 110, may determine what operation is performed.

[0103] FIGS. 8A-8C illustrates another embodiment of device 100 and its output in response to receiving input indicating a request to launch an application. FIG. 8A illustrates device 100 (with screens 104 and 108) when no applications have been launched and a desktop is displayed on both display 110 and display 114. As shown in FIG. 8B, a user can make a gesture 808, such as a tap, drag, or flick gesture on display 110 to indicate a request to launch an application. In the embodiment shown in Fig. 8B, the gesture has been made on a displayed image 808 (e.g., an icon representing the application). In some embodiments, launching of the application may be effected by two gestures, such as selecting the image 808 and then inputting a gesture in gesture capture region 120 (e.g., gesture 808A).

[0104] As show in FIG. 8B, the gesture 804 is received by touch sensitive display 114, and in response to receiving the gesture 804, an application is launched. As shown in FIG 8C, the launching of the application results in an image 812 being displayed on display 110. The image 812 is for example an open window that provides a user interface. The window will have a view of the application and allow the user to interact with the launched application. As shown in FIG. 8C, the image 812 is displayed on display 114 in response to the input, i.e. gesture 804 having been received by display 114.

[0105] In other embodiments, screen 104 may be considered the primary screen. In some of these embodiments, even if gesture 804 is received on display 114, the image 812 will be displayed on display 110 of the primary screen 104. That is, whenever an application is launched, any image that is displayed as a result of launching the application will always be displayed on a display of the primary screen, e.g., display 110 of primary screen 104.

[0106] FIGS. 9A-9C illustrates another embodiment of device 100 and its output in response to receiving inputs indicating requests to close applications and launch applications. FIG. 9A illustrates device 100 (with screens 104 and 108) when two applications have been launched. A window 904 of the first application is displayed on display 110 and comprises a view 1.1 of the first application. A user can interact with the first application using window 904. Displayed on display 114 is a second window 908 that comprises a view 2.2 of a second application that allows a user to interact the second application.

[0107] As can be appreciated, applications may have a number of different views. In embodiments, an application can display a window with a default view when launched. A user can navigate to other views. For example, a user may expand a menu and make a selection from the menu, in which case the application will display a view that reflects the user's selection. For instance, in word processing applications, a user may navigate to views that allow a user to print a document, email a document, change fonts, create a new document, save a document etc. FIG. 10 illustrates a tree 1000 showing views for an example application. Tree 1000 begins with the

main view 1004. This may be for example a default view that is displayed when the application is first launched, i.e., a main view or a home view.

[0108] Tree 1000 may be navigated when a user makes different selections or inputs into the application. A user may make a selection or input that navigates to view 1.1 (1008) or view 1.2 (1012). Other views may further depend upon views 1.1 and/or view 1.2. As one example, from view 1.1 a user may navigate to views 1.1.1., 1.1.2, or 1.1.3 (1016-1024 on tree 1000). From view 1.2 (1012) a user can navigate to view 1.2.1 (1028) and then to view 1.2.1.1 (1032), as shown in FIG. 10. As can be appreciated tree 1000 is provided for illustrative purposes only and applications may have any number of views that have a tree with more leaves and branches than shown on tree 1000.

[0109] In one embodiment, a user may make a first gesture that launches a first application and a second gesture that launches a second application. As part of interacting with the first application, a user may navigate to view 1.1, which as shown in FIG. 9A is displayed on touch sensitive display 110. The user may also interact with the second application and navigate to view 2.2, which is shown in FIG. 9A as displayed on a touch sensitive display 114.

[0110] At some point in time, a user provides an input 912 that closes the first application and/or the second application and consequently windows 904 and 908. The input 912 may be for example a gesture received by one of display 110, display 114, or some other portion of screen 104 and/or screen 108. It should be noted that although input 912 is shown in FIG. 9A as a single input, in other embodiments, more than one input is received in order to close both applications and windows 904 and 908. For example, a first gesture may be received to close window 904/the first application, and a second gesture may be received close window 908/the second application.

[0111] FIG. 9B illustrates device 100 after both windows 904 and 908 have been closed. Only desktops are shown as displayed on displays 110 and 114. Similar to FIG. 8B, a user makes a gesture 804 on display 114. As described above with respect to FIG. 8B and 8C, the gesture 804 indicates a request to open the first application. In response, a window 916 is displayed on display 114. As can be appreciated, because gesture 804 is received by display 114, when the first application is relaunched, it will display window 916 on display 114.

[0112] As shown in FIG. 9C, the window 916 shows view 1.1, which is the last view to which a user navigated in the first application before the first application was closed. In other words,

when the first application is relaunched, the view displayed to a user will be the same as the last view the user saw before closing the first application. As can be appreciated, this allows a user to continue from a point within the first application where the user left off. In some embodiments, this will save the user time because the user will not have to start from a main view and waste time navigating to the previous view.

[0113] In the embodiments in which an application that is relaunched displays a window with the last view provided to a user before closing the application, device 100, namely software within device 100, will save the view state of an application after it is closed. That is, when input 912 is received from a user, device 100 will store information regarding the current view, e.g., the view state, of the first application and the second application so that when the applications are relaunched the previous views can be displayed to a user.

[0114] Referring now to FIGS. 11A-11C, another embodiment of device 100 is show in which a multi-screen application is closed and launched. FIG. 11A illustrates that a window 1104 of a multi-screen application is displayed across both display 110 and display 114. As indicated above, device 100 is capable of various display modes and the mode illustrated in FIG. 11A may be referred to as Pmax configuration.

[0115] At some point, device 100 receives input 1108 from the user indicating a request to close the multi-screen application and window 1104. As a result, window 1104 is closed and as shown in FIG 11B displays 110 and 114 show a desktop. It is noted that in some embodiments, there will be other windows in the windows stack for one or both displays 110 and 114, in which case instead of the desktop, the displays 110 and 114 will display the next window in their stack.

[0116] A user may then make a gesture 704 on display 110 indicating a request to launch the multi-screen application. In response to gesture 704, device 100 will launch the multi-screen application, which will in turn display window 1112 on display 110. In this embodiment, window 1112 is only displayed across display 110 and not across both displays 110 and 114. A user may enter a gesture in device 100 so that window 1112 is displayed across both displays 110 and 114, but as a default position whenever the multi-screen application is opened and window 1112 displayed, it will occupies only one display, namely the display which received the gesture 704. It should be noted that in some embodiments, one of screens 104 or 108 will be designated as a primary screen. In these environments, any multi-screen application that is

launched will display a window, e.g., window 1112, in the display that corresponds to the primary screen, which may be in embodiments screen 114.

[0117] Referring now to FIGS. 12 and 13, flow diagrams 1200 and 1300 in accordance with at least some embodiments of the present disclosure are shown and will be described. Flows 1200 and 1300 are in embodiments performed by a device such as device 100. More specifically, one or more hardware or software components may be involved in performing flows 1200 and 1300. In one embodiment, modules in middleware 520 (FIG. 5A) such as multi-display management class 524 (FIG. 5B) perform one or more of the steps of flows 1200 and 1300. In other embodiments, in addition to, or in lieu of, middleware 520 performing steps of flows 1200 and 1300, operating system kernel(s) 516a, processor 504, and/or drivers 512a-512b may also perform steps of flows 1200 and 1300.

[0118] Flow 1200 is initiated at 1204. Flow 1200 passes from 1204 to optional step 1208 where input regarding launch behavior is received. In this step, a user may provide configuration information that indicates how a device should handle gestures that indicate a request to launch an application. In embodiments, the device is device 100 (FIGS. 1-3), which includes multiple screens. The user may input any information that affects how a device 100 may handle input gestures, including but not limited to, what applications are launched, changes (if any) that occur to displays 104 and 108 when a gesture has been received, what is highlighted when focus is changed, priority list of applications when a gesture is received, whether focus changes are limited to a display, screen, and/or displayed image, whether the type of gesture received results in different operations, and other information. In one embodiment, the input indicates that in response to a gesture, an application should be launched and focus should be placed on an image displayed as a result of the launching of the application.

[0119] From step 1208, flow 1200 passes to step 1212 where input is received indicating a request to launch an application. The input may be a gesture that is received in some embodiments by pressure sensitive displays 110 and 114 or gesture capture regions such as region 120. In some embodiments, as shown in FIGS. 1 and 2, the gesture capture region(s) is distinct and located away from the pressure sensitive displays. In other embodiments the gesture capture area may overlap partially or completely with the pressure sensitive displays.

[0120] The gesture may be any type of gesture. In embodiments, the gesture is one or more of the gestures illustrated in FIG. 4A-4H. The device 100 is configured so that the gesture received

at 1212 indicates a request to launch an application and change focus to an image (e.g., window) displayed as a result of launching the application.

[0121] Flow 1200 passes from 1212 to step 1216 where the application is launched. The launching of the application at step 1216 may include a number of substeps such as loading the application and other data from a storage device into memory. Step 1216 is then followed by step 1220 where a window of the application is displayed. In those embodiments in which the application is being launched for the first time, the window displayed at step 1220 may involve displaying a default view of the application.

[0122] After the window has been displayed at step 1220, a user may interact with the open application and navigate to different views of the application. The user may for example open up menus to perform operations within the launched application, navigating through a number of different views of the application.

[0123] At step 1224 input is received indicating a request to close the application. Similar to step 1212, the input is in embodiments a gesture that may be received by touch sensitive displays and/or gesture capture regions. As one example, the gesture may be a tap on a touch sensitive display, where the tap is made on top of a close button on the window displayed at step 1220.

[0124] Flow 1200 passes from step 1224 to optional step 1228 were a view state is saved. As indicated above, a user may navigate through a number of different views of the application once the application has been launched. At step 1228, the current view before the application is closed, is saved in storage so that it may be used when the application is relaunched. Step 1228 may involve for example saving the view state information on a local hard drive or other local computer readable storage medium. In other embodiments, device 100 may persist information on a remote server in which case, step 1228 may involve connecting with a remote server to store the view state information. At step 1230, the application and the window of the application is closed.

[0125] At step 1232, input is received to relaunch the application. The application is then relaunched at step 1236. Relaunching of the application at step 1236 may include a number of substeps such as loading the application and other data (such as the view state data saved at step 1228) from a storage device into memory. Step 1236 is then followed by step 1240 where a window of the application is displayed. The window displayed at step 1240 will show the last view of the application before the application was closed at step 1240. After step 1240, flow

1200 passes back to step 1224 were input to close the application is received. Flow 1200 then repeat steps 1228- 1240.

[0126] Flow 1300 is performed in embodiments by a device to launch and close multi-screen applications. Flow 1300 is initiated at 1304. Flow 1300 passes from 1304 to optional step 1308 where input regarding launch behavior is received. In this step, a user may provide configuration information that indicates how a device should handle gestures that indicate a request to launch a multi-view application. In embodiments, the device is device 100 (FIGS. 1-3), which includes multiple screens. The user may input any information that affects how a device 100 may handle input gestures, including but not limited to, what applications are launched, how are they displayed on the various displays, changes (if any) that occur to displays 104 and 108 when a gesture has been received, what is highlighted when focus is changed, priority list of applications when a gesture is received, whether focus changes are limited to a display, screen, and/or displayed image, whether the type of gesture received results in different operations, and other information. In one embodiment, the input indicates that in response to a gesture, an application should be launched and focus should be placed on an image displayed as a result of the launching of the application.

[0127] From step 1308, flow 1300 passes to step 1312 where input is received indicating a request to launch a multi-view application. The input may be a gesture that is received in some embodiments by pressure sensitive displays 110 and 114 or gesture capture regions such as region 120. In some embodiments, as shown in FIGS. 1 and 2, the gesture capture region(s) is distinct and located away from the pressure sensitive displays. In other embodiments the gesture capture area may overlap partially or completely with the pressure sensitive displays.

[0128] The gesture may be any type of gesture. In embodiments, the gesture is one or more of the gestures illustrated in FIG. 4A-4H. The device 100 is configured so that the gesture received at 1212 indicates a request to launch an application and change focus to an image (e.g., window) displayed as a result of launching the application.

[0129] Flow 1300 passes from 1312 to step 1316 where the application is launched. The launching of the application at step 1216 may include a number of substeps such as loading the multi-view application and other data from a storage device into memory. Step 1316 is then followed by step 1320 where a window of the application is displayed. The window will first be displayed on one display by default. However, flow 1300 may include receiving some additional

input that at step 1320 makes the window of the multi-view application be displayed on more than one display. The window may include a view of the multi-screen application that is displayed on at least a portion of a first display, e.g., display 110 and a portion of a second display, e.g., display 114.

[0130] After the window has been displayed at step 1220, a user may interact with the open application and navigate to different views of the application. The user may for example open up menus to perform operations within the launched application, navigating through a number of different views of the application.

[0131] At step 1324 input is received indicating a request to close the multi-view application. Similar to step 1312, the input is in embodiments a gesture that may be received by touch sensitive displays and/or gesture capture regions. As one example, the gesture may be a tap on a touch sensitive display, where the tap is made on top of a close button on the window displayed at step 1320.

[0132] Flow 1300 passes from step 1324 to optional step 1328 were a view state is saved. As indicated above, a user may navigate through a number of different views of the application once the application has been launched. At step 1328, the current view before the application is closed, is saved in storage so that it may be used when the application is relaunched. Step 1328 may involve for example saving the view state information on a local hard drive or other local computer readable storage medium. In other embodiments, device 100 may persist information on a remote server in which case, step 1328 may involve connecting with a remote server to store the view state information. At step 1330, the multi-view application and the window of the application is closed.

[0133] At step 1332, input is received to relaunch the application. The multi-view application is then relaunched at step 1336. Relaunching of the application at step 1336 may include a number of substeps such as loading the application and other data (such as the view state data saved at step 1328) from a storage device into memory. Step 1336 is then followed by step 1340 where a window of the application is displayed. The window displayed at step 1340 will show the last view of the application before the application was closed. However, as noted above with respect to step 1320, the window of the multi-view application will be in embodiments displayed on a single display by default. After step 1340, flow 1300 passes back to step 1324 were input to close the multi-view application is received. Flow 1300 can then repeat steps 1328- 1340.

[0134] The exemplary systems and methods of this disclosure have been described in relation to FIGS 1-10. However, to avoid unnecessarily obscuring the present disclosure, the preceding description omits a number of known structures and devices. This omission is not to be construed as a limitation of the scopes of the claims. Specific details are set forth to provide an understanding of the present disclosure. It should however be appreciated that the present disclosure may be practiced in a variety of ways beyond the specific detail set forth herein.

[0135] Furthermore, while the exemplary aspects, embodiments, and/or configurations illustrated herein show the various components of the system collocated, certain components of the system can be located remotely, at distant portions of a distributed network, such as a LAN and/or the Internet, or within a dedicated system. Thus, it should be appreciated, that the components of the system can be combined in to one or more devices, such as a phone, computer, PDA, electronic book reader, gaming device, or collocated on a particular node of a distributed network, such as an analog and/or digital telecommunications network, a packet-switch network, or a circuit-switched network. It will be appreciated from the preceding description, and for reasons of computational efficiency, that the components of the system can be arranged at any location within a distributed network of components without affecting the operation of the system. For example, the various components can be located in a switch such as a PBX and media server, gateway, in one or more communications devices, at one or more users' premises, or some combination thereof. Similarly, one or more functional portions of the system could be distributed between a telecommunications device(s) and an associated computing device.

[0136] Furthermore, it should be appreciated that the various links connecting the elements can be wired or wireless links, or any combination thereof, or any other known or later developed element(s) that is capable of supplying and/or communicating data to and from the connected elements. These wired or wireless links can also be secure links and may be capable of communicating encrypted information. Transmission media used as links, for example, can be any suitable carrier for electrical signals, including coaxial cables, copper wire and fiber optics, and may take the form of acoustic or light waves, such as those generated during radio-wave and infra-red data communications.

[0137] Also, while the flowcharts have been discussed and illustrated in relation to a particular sequence of events, it should be appreciated that changes, additions, and omissions to this

sequence can occur without materially affecting the operation of the disclosed embodiments, configuration, and aspects.

[0138] A number of variations and modifications of the disclosure can be used. It would be possible to provide for some features of the disclosure without providing others.

[0139] For example in one alternative embodiment, all of the focus behavior information and information regarding launching applications may be preprogrammed into the device 100. In these embodiments, there would be no input received from a user regarding how a device 100 handles gestures indicating the launching of applications or a change in focus.

[0140] In other alternative embodiments, device 100 may be in any configuration described above with respect to FIGS. 6A-6J. In these embodiments, gestures would be received as described above indicating a request for launching an application or changing focus.

[0141] In yet another embodiment, the systems and methods of this disclosure can be implemented in conjunction with a special purpose computer, a programmed microprocessor or microcontroller and peripheral integrated circuit element(s), an ASIC or other integrated circuit, a digital signal processor, a hard-wired electronic or logic circuit such as discrete element circuit, a programmable logic device or gate array such as PLD, PLA, FPGA, PAL, special purpose computer, any comparable means, or the like. In general, any device(s) or means capable of implementing the methodology illustrated herein can be used to implement the various aspects of this disclosure. Exemplary hardware that can be used for the disclosed embodiments, configurations and aspects includes computers, handheld devices, telephones (e.g., cellular, Internet enabled, digital, analog, hybrids, and others), and other hardware known in the art. Some of these devices include processors (e.g., a single or multiple microprocessors), memory, nonvolatile storage, input devices, and output devices. Furthermore, alternative software implementations including, but not limited to, distributed processing or component/object distributed processing, parallel processing, or virtual machine processing can also be constructed to implement the methods described herein.

[0142] In yet another embodiment, the disclosed methods may be readily implemented in conjunction with software using object or object-oriented software development environments that provide portable source code that can be used on a variety of computer or workstation platforms. Alternatively, the disclosed system may be implemented partially or fully in hardware using standard logic circuits or VLSI design. Whether software or hardware is used to

implement the systems in accordance with this disclosure is dependent on the speed and/or efficiency requirements of the system, the particular function, and the particular software or hardware systems or microprocessor or microcomputer systems being utilized.

[0143] In yet another embodiment, the disclosed methods may be partially implemented in software that can be stored on a storage medium, executed on programmed general-purpose computer with the cooperation of a controller and memory, a special purpose computer, a microprocessor, or the like. In these instances, the systems and methods of this disclosure can be implemented as program embedded on personal computer such as an applet, JAVA® or CGI script, as a resource residing on a server or computer workstation, as a routine embedded in a dedicated measurement system, system component, or the like. The system can also be implemented by physically incorporating the system and/or method into a software and/or hardware system.

[0144] Although the present disclosure describes components and functions implemented in the aspects, embodiments, and/or configurations with reference to particular standards and protocols, the aspects, embodiments, and/or configurations are not limited to such standards and protocols. Other similar standards and protocols not mentioned herein are in existence and are considered to be included in the present disclosure. Moreover, the standards and protocols mentioned herein and other similar standards and protocols not mentioned herein are periodically superseded by faster or more effective equivalents having essentially the same functions. Such replacement standards and protocols having the same functions are considered equivalents included in the present disclosure.

[0145] The present disclosure, in various aspects, embodiments, and/or configurations, includes components, methods, processes, systems and/or apparatus substantially as depicted and described herein, including various aspects, embodiments, configurations embodiments, subcombinations, and/or subsets thereof. Those of skill in the art will understand how to make and use the disclosed aspects, embodiments, and/or configurations afrer understanding the present disclosure. The present disclosure, in various aspects, embodiments, and/or configurations, includes providing devices and processes in the absence of items not depicted and/or described herein or in various aspects, embodiments, and/or configurations hereof, including in the absence of such items as may have been used in previous devices or processes, e.g., for improving performance, achieving ease and\or reducing cost of implementation.

[0146] The foregoing discussion has been presented for purposes of illustration and description. The foregoing is not intended to limit the disclosure to the form or forms disclosed herein. In the foregoing Detailed Description for example, various features of the disclosure are grouped together in one or more aspects, embodiments, and/or configurations for the purpose of streamlining the disclosure. The features of the aspects, embodiments, and/or configurations of the disclosure may be combined in alternate aspects, embodiments, and/or configurations other than those discussed above. This method of disclosure is not to be interpreted as reflecting an intention that the claims require more features than are expressly recited in each claim. Rather, as the following claims reflect, inventive aspects lie in less than all features of a single foregoing disclosed aspect, embodiment, and/or configuration. Thus, the following claims are hereby incorporated into this Detailed Description, with each claim standing on its own as a separate preferred embodiment of the disclosure.

[0147] Moreover, though the description has included description of one or more aspects, embodiments, and/or configurations and certain variations and modifications, other variations, combinations, and modifications are within the scope of the disclosure, e.g., as may be within the skill and knowledge of those in the art, afrer understanding the present disclosure. It is intended to obtain rights which include alternative aspects, embodiments, and/or configurations to the extent permitted, including alternate, interchangeable and/or equivalent structures, functions, ranges or steps to those claimed, whether or not such alternate, interchangeable and/or equivalent structures, functions, ranges or steps are disclosed herein, and without intending to publicly dedicate any patentable subject matter.

What is claimed is:

1. A method, comprising:

receiving first input indicating a request to launch a first application, wherein the input is received by a first touch sensitive display of a first screen;

launching the first application;

the first application, displaying a first image on the first touch sensitive display of the first screen;

receiving second input indicating a request to launch a second application, wherein the input is received by a second touch sensitive display of a second screen;

launching the second application; and

the second application, displaying a second image on the second touch sensitive display of the second screen.

2. The method of claim 1, wherein the first image is a first open window of the first application and the second image is a second open window of the second application.

3. The method of claim 1, wherein the first image is a last view of the first application before the first application was closed.

4. The method of claim 1, wherein the second image is a last view of the second application before the second application was closed.

5. The method of claim 1, wherein the first input is a tap gesture.

6. The method of claim 1, wherein the second input is a tap gesture.

7. The method of claim 1, wherein the first image is not a main view of the first application.

.

8. The method of claim 1, wherein the second image is not a main view of the second application.

9. The method of claim 1, wherein the first input is a tap gesture on an icon that represents the first application.

10. A non-transitory computer readable medium storing computer executable instructions that when executed by at least one processor perform a method comprising:

receiving first input indicating a request to close an application, the application displaying a first window with a first view of the application, wherein the first window is displayed on one of a first touch screen display of a first screen or a second touch screen display of a second screen;

in response to receiving the input, closing the first window;

receiving second input indicating a request to launch the application;

launching the application; and

in response to receiving the input indicating a request to launch the application, displaying a second window with the first view of the application.

11. The non-transitory computer readable medium of claim 10, wherein the second window is displayed on the one of the first touch screen display of the first screen or the second touch screen display of the second screen.

12. The non-transitory computer readable medium of claim 10, wherein the first view of the application is a leaf view of the application.

13. The non-transitory computer readable medium of claim 10, wherein the first window is displayed on the first touch screen display, and the second window is displayed on the second touch screen display.

14. The non-transitory computer readable medium of claim 13, wherein the second input is received by the second touch sensitive display, and in response to the second touch

sensitive display receiving the second input, the second window being displayed on the second touch sensitive display.

15. The non-transitory computer readable medium of claim 10, wherein the wherein the second input comprises a tap on the second touch sensitive display.

16. The non-transitory computer readable medium of claim 15, wherein the tap is on a displayed icon of the second application.

17. A dual screen communication device, comprising:

a first touch sensitive display of a first screen;

a second touch sensitive display of a second screen;

a computer readable medium that stores computer executable instructions that when executed by at least one processor perform a method comprising:

displaying a window with a view of a multi-screen application, wherein the window is displayed on at least a first portion of a first touch sensitive display of a first screen and at least a second portion of a second touch sensitive display of a second screen;

receiving first input indicating a request to close the multi-screen application;

closing the window of the multi-screen application;

receiving second input indicating a request to launch the multi-screen application;

launching the multi-screen application; and

displaying at least a portion of the view in a second window on one of the first touch sensitive display or the second touch sensitive display.

18. The device of claim 17, wherein the second input was received by the first touch sensitive display and, in response, the second window being displayed in the first touch sensitive display.

19. The device of claim 17, wherein the method further comprises:

wherein the second input was received by the second touch sensitive display and wherein the second window is displayed in the first touch sensitive display.

20. The device of claim 17, wherein not all of the view is displayed in the second window.

ABSTRACT

Embodiments are described for handling the launching of applications in a multi-screen device. In embodiments, a first touch sensitive display of a first screen receives input to launch an application. In response, the application is launched and a window of the first application is displayed on the first display. A second touch sensitive display of a second screen receives input to launch a second application. In response, the second application is launched and a second window of the second application is displayed on the second display. In embodiments, when an application is launched, it displays the view of the application (whether on the first touch sensitive display or the second touch sensitive display) that was displayed when the application was last closed.

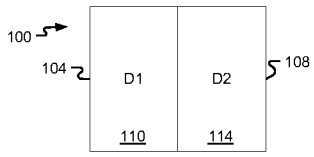


FIG. 7A

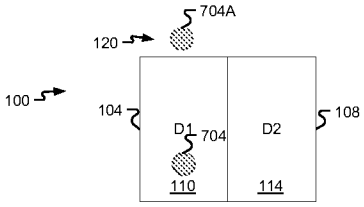


FIG. 7B

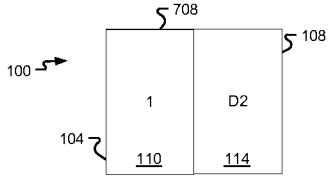


FIG. 7C

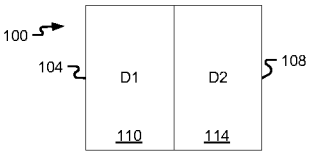


FIG. 8A

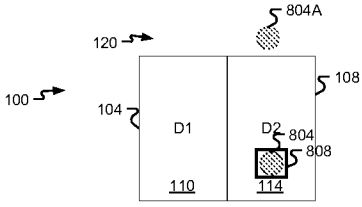


FIG. 8B

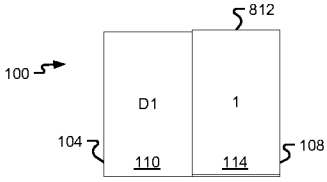


FIG. 8C

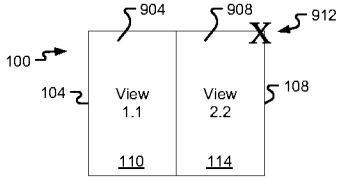


FIG. 9A

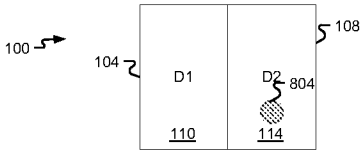


FIG. 9B

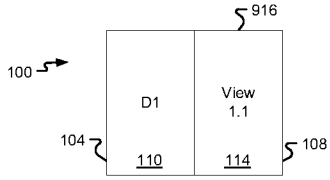


FIG. 9C

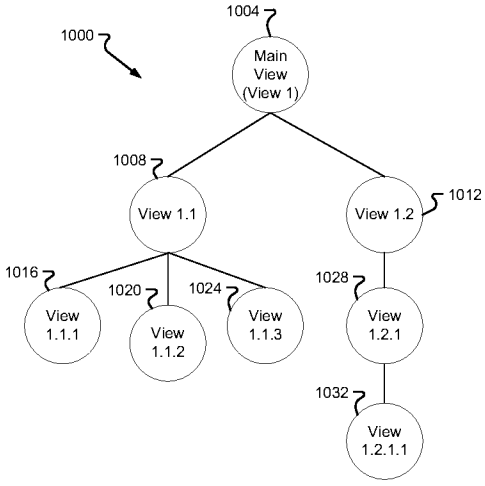


FIG. 10

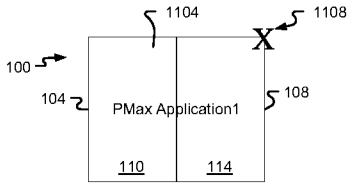


FIG. 11A

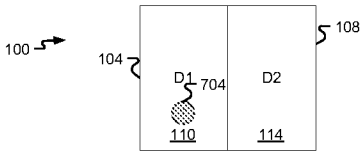


FIG. 11B

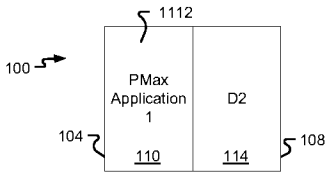


FIG. 11C

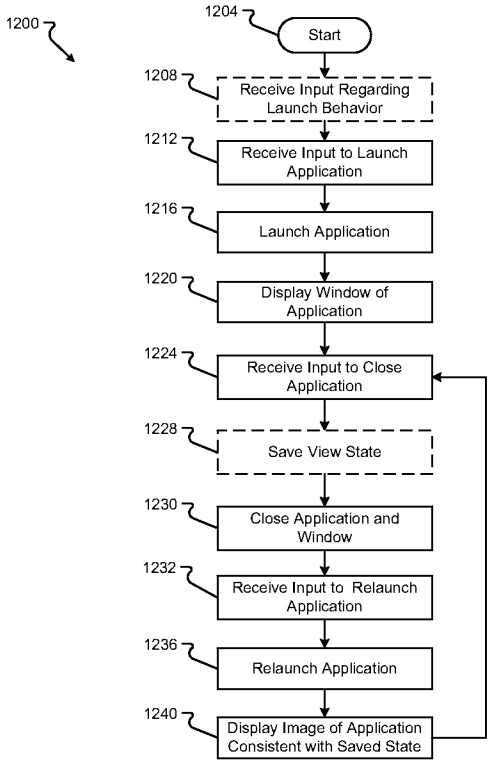


FIG. 12

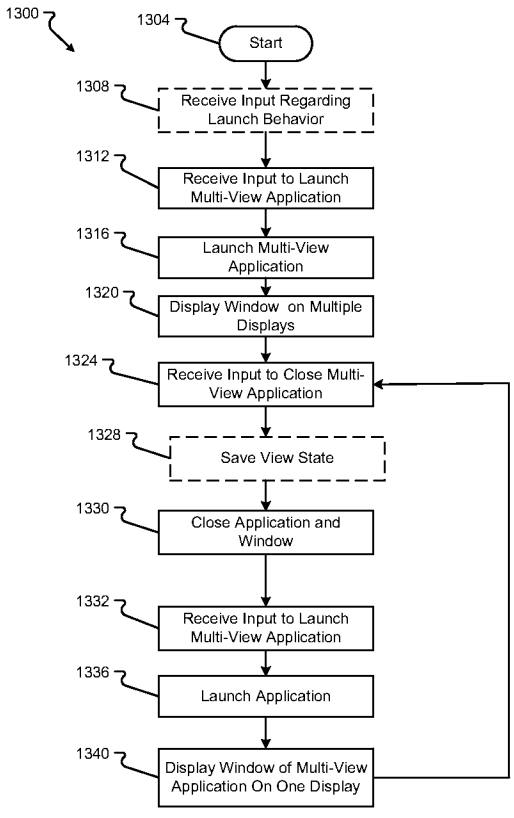


FIG. 13

[0100] FIGS. 7A-7C illustrate an embodiment showing device 100 and how device 100 responds when applications are launched. FIG. 7A illustrates device 100 (with screens 104 and 108) when no applications have been launched and a desktop is displayed on both display 110 and display 114. As shown in FIG. 7B, a user can make a gesture 708, such as a tap, drag, or flick gesture on display 110 to indicate a request to launch an application. In some embodiments, the gesture may be made in gesture capture region 120 (e.g., gesture 708A). Device 100 receives the gesture 708 or 708A. In response to receiving the gesture, device 100 will launch the application. As shown in FIG 7C, the launching of the application results in an image 708 being displayed on display 110. The image 708 is for example an open window that provides a user interface. The window will have a view of the application and allow the user to interact with the launched application. As shown in FIG. 7C, the image 708 is displayed on display 110 because the input, i.e. gesture 704 was received on display 110, and there were no windows displayed on display 110 or in the stack of display 110.

[0101] It is noted that gesture 704 may be made on one or more portions of display 110, a configurable area of screen 104, or as shown in FIG 7B (gesture 708A) a gesture capture region 120. The gesture 708 may involve touching portions of screen 104 and or can involve only movement above the surface of portions of screen 104. In some embodiments, for the gesture to be recognized, contact is required to be made with the display 110 for a predetermined period of time.

[0102] In some embodiments, device 100 is configured to interpret different gestures as indications of different operations. For example, the device 100 may be configured to move an image when the gesture 708 is a drag gesture or flick gesture, but launch applications if gesture 708 is a flick, tap gesture, or double tap gesture. In embodiments, when an application is launched, any image displayed in response to the launching of the application (e.g., open window) is placed in focus. This allows a user to immediately begin to interact with the open application to the displayed image. In some embodiments, the location of the gesture 708, whether in gesture capture region 120 or some other portion of display 110, may determine what operation is performed.

[0103] FIGS. 8A-8C illustrates another embodiment of device 100 and its output in response to receiving input indicating a request to launch an application. FIG. 8A illustrates device 100 (with screens 104 and 108) when no applications have been launched and a desktop is displayed on both display 110 and display 114. As shown in FIG. 8B, a user can make a gesture 808, such as a tap, drag, or flick gesture on display 110 to indicate a request to launch an application. In the embodiment shown in Fig. 8B, the gesture has been made on a displayed image 808 (e.g., an icon representing the application). In some embodiments, launching of the application may be effected by two gestures, such as selecting the image 808 and then inputting a gesture in gesture capture region 120 (e.g., gesture 808A).

[0104] As show in FIG. 8B, the gesture 804 is received by touch sensitive display 114, and in response to receiving the gesture 804, an application is launched. As shown in FIG 8C, the launching of the application results in an image 812 being displayed on display 110. The image 812 is for example an open window that provides a user interface. The window will have a view of the application and allow the user to interact with the launched application. As shown in FIG. 8C, the image 812 is displayed on display 114 in response to the input, i.e. gesture 804 having been received by display 114, and there were no windows displayed on display 114 or in the stack of display 114.

[0105] In other embodiments, screen 104 may be considered the primary screen. In some of these embodiments, even if gesture 804 is received on display 114, the image 812 will be displayed on display 110 of the primary screen 104. That is, whenever an application is launched, any image that is displayed as a result of launching the application will always be displayed on a display of the primary screen, e.g., display 110 of primary screen 104, even if the display is already displaying windows, i.e., has windows in its stack.

[0106] FIGS. 9A-9C illustrates another embodiment of device 100 and its output in response to receiving inputs indicating requests to launch applications. FIG. 9A illustrates device 100 (with screens 104 and 108) when a first application has been launched. A window 904 of the first application is displayed on display 110. A user can interact with the first application using window 904. In FIG. 9A, no windows are displayed on display 114.

[0107] As shown in FIG. 9B, at some point in time, a user provides an input 908 that indicates a request to open a second application. The input 908 may be for example a gesture received by gesture capture region 120, or one of display 110, display 114, or some other portion of screen 104 and/or screen 108. It should be noted that although input 908 is shown in FIG. 9A as a single input, in other embodiments, more than one input is received in order to indicate a request

to launch an application, such as selection of an icon and a gesture indicating a request to open the application represented by the icon.

[0108] As shown in FIG. 9C, in response to the gesture 908, a second application is launched and the second application displays window 912 on display 114. In this embodiment, the window 912 is displayed on display 114 because display 114 is not already displaying other windows. That is, there are no windows in the stack of display 114, but there is a window displayed on display 110. In these embodiments, device 100 operates by opening new windows on a display that is not currently displaying any other windows. This is true even if the input to open the second application is received by display 110. In other words, if the user opens up the second application by making the gesture on display 110, window 912 will still be displayed on 114 because it is not being occupied by another window.

[0109] Similarly, if a window is displayed on display 114, when any application is launched and a new application window is displayed, it will be displayed on display 110 if it is not currently displaying any other windows. However, in some embodiments, if both displays are currently displaying windows and device 100 receives input to open a new application a default display may be selected, (a designated primary display). In other embodiments, the display that received the input will be used to display the new window of the newly launched application.

[0110] Referring now to FIG. 10, flow diagram 1000 in accordance with at least some embodiments of the present disclosure is shown and will be described. Flow 1000 is in embodiments performed by a device such as device 100. More specifically, one or more hardware or software components may be involved in performing flow 1000. In one embodiment, modules in middleware 520 (FIG. 5A) such as multi-display management class 524 (FIG. 5B) perform one or more of the steps of flow 1000. In other embodiments, in addition to, or in lieu of, middleware 520 performing steps of flows 1000, operating system kernel(s) 516a, processor 504, and/or drivers 512a-512b may also perform steps of flow 1000.

[0111] Flow 1000 is initiated at 1004. Flow 1000 passes from 1004 to optional step 1008 where input regarding launch behavior is received. In this step, a user may provide configuration information that indicates how a device should handle gestures that indicate a request to launch an application. In embodiments, the device is device 100 (FIGS. 1-3), which includes multiple screens. The user may input any information that affects how a device 100 may handle input gestures, including but not limited to, what applications are launched, changes (if any) that occur

to displays 104 and 108 when a gesture has been received, what is highlighted when focus is changed, priority list of applications when a gesture is received, whether focus changes are limited to a display, screen, and/or displayed image, whether the type of gesture received results in different operations, where windows of newly launched applications are displayed, and other information. In one embodiment, the input indicates that in response to a gesture, an application should be launched and the window should be displayed on a specific display.

[0112] From step 1008, flow 1000 passes to step 1012 where input is received indicating a request to launch an application. The input may be a gesture that is received in some embodiments by pressure sensitive displays 110 and 114 or gesture capture regions such as region 120. In some embodiments, as shown in FIGS. 1 and 2, the gesture capture region(s) is distinct and located away from the pressure sensitive displays. In other embodiments the gesture capture area may overlap partially or completely with the pressure sensitive displays.

[0113] The gesture may be any type of gesture. In embodiments, the gesture is one or more of the gestures illustrated in FIG. 4A-4H. The device 100 is configured so that the gesture received at 1012 indicates a request to launch an application and change focus to an image (e.g., window) displayed as a result of launching the application.

[0114] Flow 1000 passes from step 1012 to step 1014 where the application is launched. The launching of the application at step 1014 may include a number of substeps such as loading the application and other data from a storage device into memory.

[0115] Following step 1014, flow passes to decision 1016 where a determination is made whether the display that received the input at step 1012 has an empty stack. As can be appreciated, in some embodiments each display (110 and 114) of device 100 may have a separate stack of windows that are displayed depending on what window is selected by a user. In other embodiments, device 100 may have a single stack for both displays, with the stack being moved from one display to another. At decision 1016, a determination is made as to whether the display that received the input at step 1012 has an empty stack, i.e., there are no other windows displayed on the display that received the input.

[0116] If at decision 1016, a determination is made that the display that received the input does have an empty stack, then the launched application will display a window and the display that received the input. Flow 1000 then ends at 1024. If at decision 1016 a determination is made

that the display that received the input does not have an empty stack, flow will pass to decision 1028, where a decision is made as to whether or not the second display has an empty stack.

[0117] If at decision 1028, a determination is made that the second display does have an empty stack, the window of the launched application will be displayed on the second display, which has the empty stack. This feature allows the device to take advantage of the fact that one of the displays is not currently being used and can therefore be utilized to display the window of the newly launched application.

[0118] If at decision 1028, a determination is made that the second display does not have an empty stack, then flow 1000 passes to step 1020 where the window of the newly launched application is displayed on the display that received the input. Because under these circumstances, both displays are currently displaying windows or have windows in their stack, a default position for displaying the window of the newly launched application is the display that received the input. Indeed, a user may expect that since the input was made on a particular display, the window of the newly launched application should be displayed on the same display that received the input.

[0119] The exemplary systems and methods of this disclosure have been described in relation to FIGS 1-10. However, to avoid unnecessarily obscuring the present disclosure, the preceding description omits a number of known structures and devices. This omission is not to be construed as a limitation of the scopes of the claims. Specific details are set forth to provide an understanding of the present disclosure. It should however be appreciated that the present disclosure may be practiced in a variety of ways beyond the specific detail set forth herein.

[0120] Furthermore, while the exemplary aspects, embodiments, and/or configurations illustrated herein show the various components of the system collocated, certain components of the system can be located remotely, at distant portions of a distributed network, such as a LAN and/or the Internet, or within a dedicated system. Thus, it should be appreciated, that the components of the system can be combined in to one or more devices, such as a phone, computer, PDA, electronic book reader, gaming device, or collocated on a particular node of a distributed network, such as an analog and/or digital telecommunications network, a packet-switch network, or a circuit-switched network. It will be appreciated from the preceding description, and for reasons of computational efficiency, that the components of the system can be arranged at any location within a distributed network of components without affecting the

operation of the system. For example, the various components can be located in a switch such as a PBX and media server, gateway, in one or more communications devices, at one or more users' premises, or some combination thereof. Similarly, one or more functional portions of the system could be distributed between a telecommunications device(s) and an associated computing device.

[0121] Furthermore, it should be appreciated that the various links connecting the elements can be wired or wireless links, or any combination thereof, or any other known or later developed element(s) that is capable of supplying and/or communicating data to and from the connected elements. These wired or wireless links can also be secure links and may be capable of communicating encrypted information. Transmission media used as links, for example, can be any suitable carrier for electrical signals, including coaxial cables, copper wire and fiber optics, and may take the form of acoustic or light waves, such as those generated during radio-wave and infra-red data communications.

[0122] Also, while the flowcharts have been discussed and illustrated in relation to a particular sequence of events, it should be appreciated that changes, additions, and omissions to this sequence can occur without materially affecting the operation of the disclosed embodiments, configuration, and aspects.

[0123] A number of variations and modifications of the disclosure can be used. It would be possible to provide for some features of the disclosure without providing others.

[0124] For example in one alternative embodiment, all of the focus behavior information and information regarding launching applications may be preprogrammed into the device 100. In these embodiments, there would be no input received from a user regarding how a device 100 handles gestures indicating the launching of applications or a change in focus.

[0125] In other alternative embodiments, device 100 may be in any configuration described above with respect to FIGS. 6A-6J. In these embodiments, gestures would be received as described above indicating a request for launching an application or changing focus.

[0126] In yet another embodiment, the systems and methods of this disclosure can be implemented in conjunction with a special purpose computer, a programmed microprocessor or microcontroller and peripheral integrated circuit element(s), an ASIC or other integrated circuit, a digital signal processor, a hard-wired electronic or logic circuit such as discrete element circuit, a programmable logic device or gate array such as PLD, PLA, FPGA, PAL, special purpose

computer, any comparable means, or the like. In general, any device(s) or means capable of implementing the methodology illustrated herein can be used to implement the various aspects of this disclosure. Exemplary hardware that can be used for the disclosed embodiments, configurations and aspects includes computers, handheld devices, telephones (e.g., cellular, Internet enabled, digital, analog, hybrids, and others), and other hardware known in the art. Some of these devices include processors (e.g., a single or multiple microprocessors), memory, nonvolatile storage, input devices, and output devices. Furthermore, alternative software implementations including, but not limited to, distributed processing or component/object distributed processing, parallel processing, or virtual machine processing can also be constructed to implement the methods described herein.

[0127] In yet another embodiment, the disclosed methods may be readily implemented in conjunction with software using object or object-oriented software development environments that provide portable source code that can be used on a variety of computer or workstation platforms. Alternatively, the disclosed system may be implemented partially or fully in hardware using standard logic circuits or VLSI design. Whether software or hardware is used to implement the systems in accordance with this disclosure is dependent on the speed and/or efficiency requirements of the system, the particular function, and the particular software or hardware systems or microprocessor or microcomputer systems being utilized.

[0128] In yet another embodiment, the disclosed methods may be partially implemented in software that can be stored on a storage medium, executed on programmed general-purpose computer with the cooperation of a controller and memory, a special purpose computer, a microprocessor, or the like. In these instances, the systems and methods of this disclosure can be implemented as program embedded on personal computer such as an applet, JAVA® or CGI script, as a resource residing on a server or computer workstation, as a routine embedded in a dedicated measurement system, system component, or the like. The system can also be implemented by physically incorporating the system and/or method into a software and/or hardware system.

[0129] Although the present disclosure describes components and functions implemented in the aspects, embodiments, and/or configurations with reference to particular standards and protocols, the aspects, embodiments, and/or configurations are not limited to such standards and protocols. Other similar standards and protocols not mentioned herein are in existence and are

considered to be included in the present disclosure. Moreover, the standards and protocols mentioned herein and other similar standards and protocols not mentioned herein are periodically superseded by faster or more effective equivalents having essentially the same functions. Such replacement standards and protocols having the same functions are considered equivalents included in the present disclosure.

[0130] The present disclosure, in various aspects, embodiments, and/or configurations, includes components, methods, processes, systems and/or apparatus substantially as depicted and described herein, including various aspects, embodiments, configurations embodiments, subcombinations, and/or subsets thereof. Those of skill in the art will understand how to make and use the disclosed aspects, embodiments, and/or configurations afrer understanding the present disclosure. The present disclosure, in various aspects, embodiments, and/or configurations, includes providing devices and processes in the absence of items not depicted and/or described herein or in various aspects, embodiments, and/or configurations hereof, including in the absence of such items as may have been used in previous devices or processes, e.g., for improving performance, achieving ease and\or reducing cost of implementation.

[0131] The foregoing discussion has been presented for purposes of illustration and description. The foregoing is not intended to limit the disclosure to the form or forms disclosed herein. In the foregoing Detailed Description for example, various features of the disclosure are grouped together in one or more aspects, embodiments, and/or configurations for the purpose of streamlining the disclosure. The features of the aspects, embodiments, and/or configurations of the disclosure may be combined in alternate aspects, embodiments, and/or configurations other than those discussed above. This method of disclosure is not to be interpreted as reflecting an intention that the claims require more features than are expressly recited in each claim. Rather, as the following claims reflect, inventive aspects lie in less than all features of a single foregoing disclosed aspect, embodiment, and/or configuration. Thus, the following claims are hereby incorporated into this Detailed Description, with each claim standing on its own as a separate preferred embodiment of the disclosure.

[0132] Moreover, though the description has included description of one or more aspects, embodiments, and/or configurations and certain variations and modifications, other variations, combinations, and modifications are within the scope of the disclosure, e.g., as may be within the skill and knowledge of those in the art, afrer understanding the present disclosure. It is intended

to obtain rights which include alternative aspects, embodiments, and/or configurations to the extent permitted, including alternate, interchangeable and/or equivalent structures, functions, ranges or steps to those claimed, whether or not such alternate, interchangeable and/or equivalent structures, functions, ranges or steps are disclosed herein, and without intending to publicly dedicate any patentable subject matter.

What is claimed is:

1. A method, comprising:

receiving first input indicating a request to launch a first application, wherein the input is received by a first touch sensitive display of a first screen;

launching the first application;

determining whether the first touch sensitive display has an empty stack; and

in response to determining that the first touch sensitive display has an empty stack, the first application displaying a first image on the first touch sensitive display.

2. The method of claim 1, further comprising:

in response to determining that the first touch sensitive display does not have an empty stack, determining that a second touch sensitive display has an empty stack; and

the first application displaying the first image on the second touch sensitive display.

3. The method of claim 1, further comprising:

in response to determining that the first touch sensitive display does not have an empty stack, determining that a second touch sensitive display also does not have an empty stack; and

the first application displaying the first image on the first touch sensitive display.

4. The method of claim 1, further comprising:

receiving second input indicating a request to launch a second application, wherein the input is received by a second touch sensitive display of a second screen;

launching the second application;

determining that the second touch sensitive display has an empty stack;

in response to the determining that the second touch sensitive display has an empty stack, the second application, displaying a second image on the second touch sensitive display of the second screen.

5. The method of claim 1, wherein the first image is a first open window of the first application.

6. The method of claim 4, the second image is a second open window of the second application.

7. The method of claim 1, wherein the first input is a tap gesture.

8. The method of claim 4, wherein the second input is a tap gesture.

9. The method of claim 1, wherein the first input is a tap gesture on an icon that represents the first application.

10. A non-transitory computer readable medium storing computer executable instructions that when executed by at least one processor perform a method comprising:

receiving first input indicating a request to launch an application, the input received by one of a first touch screen display of a first screen or a second touch screen display of a second screen;

determining whether the one of the first touch sensitive display or the second touch sensitive display has an empty stack;

in response to the determining that the one of the first touch sensitive display or the second touch sensitive display has an empty stack, displaying an open window of the application on the one of the first touch sensitive display or the second touch sensitive display;

in response to the determining that the one of the first touch sensitive display or the second touch sensitive display does not have an empty stack, displaying the open window of the application on the other one of the first touch sensitive display or the second touch sensitive display.

11. The non-transitory computer readable medium of claim 10, wherein the other one of the first touch sensitive display or the second touch sensitive display has an empty stack.

12. The non-transitory computer readable medium of claim 10, wherein the other one of the first touch sensitive display or the second touch sensitive display does not have an empty stack and is on a screen designated a primary screen.

13. The non-transitory computer readable medium of claim 10, wherein the first input comprises a tap.

14. The non-transitory computer readable medium of claim 15, wherein the tap is on a displayed icon of the first application.

15. A dual screen communication device, comprising:

a first touch sensitive display of a first screen;

a second touch sensitive display of a second screen;

a computer readable medium that stores computer executable instructions that when executed by at least one processor perform a method comprising:

displaying a first window of a first application on the first touch sensitive display;

receiving by the first touch sensitive display, input indicating a request to launch a second application;

launching the second application;

in response to determining that no window is displayed in the second touch sensitive display, displaying a second window of the second application on the second touch sensitive display.

16. The device of claim 15, wherein the method further comprises:

receiving by the second touch sensitive display, input indicating a request to launch a third application;

launching the third application;

displaying a third window of the third application on the second touch sensitive display.

17. The device of claim 15, wherein the method further comprises:

receiving by the second touch sensitive display, input indicating a request to launch a third application;

launching the third application;

displaying a third window of the third application on the second touch sensitive display.

18. The device of claim 15, wherein the method further comprises:

receiving, by the first touch sensitive display, input indicating a request to close the first window;

closing the first window, wherein no other windows are displayed on the first touch sensitive display;

receiving by the second touch sensitive display, input indicating a request to launch a third application;

launching the third application;

displaying a third window of the third application on the first touch sensitive display.

19. The device of claim 15, wherein the method further comprises:

receiving by the first touch sensitive display, input indicating a request to launch a third application;

launching the third application;

displaying a third window of the third application on the first touch

launching the third application;

displaying a third window of the third application on the second touch sensitive display.

18. The device of claim 15, wherein the method further comprises:

receiving, by the second touch sensitive display, input indicating a request to close the second window;

closing the second window, wherein no other windows are displayed on the second touch sensitive display;

receiving by the first touch sensitive display, input indicating a request to launch a third application;

ABSTRACT

Embodiments are described for handling the launching of applications in a multi-screen device. In embodiments, a first touch sensitive display of a first screen receives input to launch an application. In response, the application is launched. A determination is made as to whether the first touch sensitive display already has windows in its stack. If there are no windows in the stack of the first touch sensitive display, a new window of the first application is displayed on the first touch sensitive display. If there are windows in the stack, a determination is made whether a second display has windows in its stack. If not, the new window is displayed on the second display. If the second display also has windows in its stack, the new window will be displayed on the first touch sensitive display.

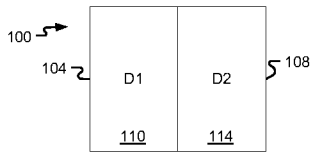


FIG. 7A

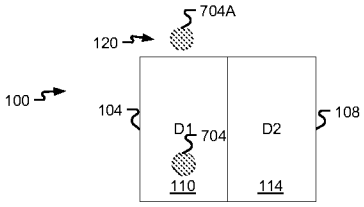


FIG. 7B

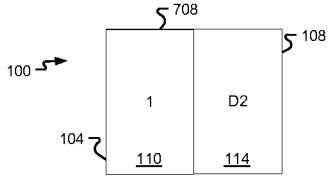


FIG. 7C

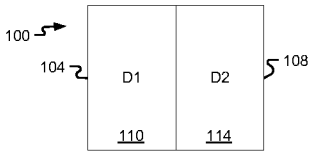


FIG. 8A

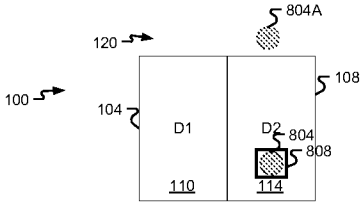


FIG. 8B

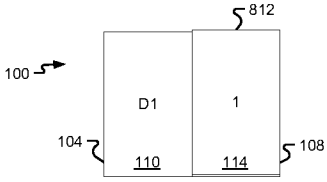


FIG. 8C

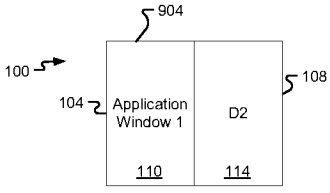


FIG. 9A

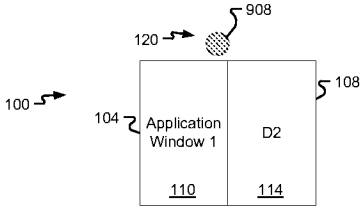


FIG. 9B

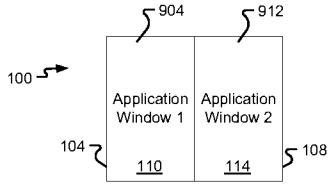


FIG. 9C

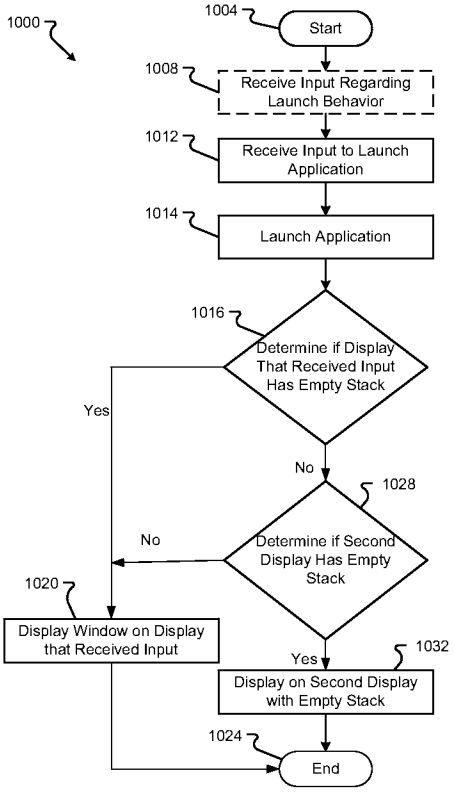


FIG. 10

[0100] FIGS. 7A-7C illustrate an embodiment showing device 100 and how device 100 responds when input is received to open a window of an application. FIG. 7A illustrates device 100 (with screens 104 and 108), with a window 704 of a multi-screen application being displayed across at least a portion of the display 110 of screen 104 and at least a portion of the display 114 of screen 108. FIG. 7A illustrates that the multi-screen application has been launched and the window displayed. As shown in FIG. 7B, input has been entered into device 100 using gesture 708 on display 114. The input indicates a request to open a second application. In response to the input, a second window 712 of the newly launched application is displayed on display 114 as shown in FIG. 7C.

[0101] The gesture 708 may be in embodiments any type of gesture such as a tap, drag, or flick gesture on display 114 to indicate a request to activate an application. Device 100 receives the gesture 708. In response to receiving the gesture, device 100 will launch the application, if the application has not been launched, or redisplay a window of the application that has already been launched. As shown in FIG 7C, the activating of the application results in a window 712 being displayed on display 114.

[0102] Because window 712, is displayed on display 114, window 704, which occupies at least a portion of display 114, cannot continue to be displayed. As a result, window 704 is minimized by displaying a minimized window 716. The minimized window 716 is displayed on display 110. In some embodiments, the minimized window 716 will include only a portion of the content displayed in windows 704. In other embodiments, window 716 may display all of the contents displayed within window 704 but reduced in size to fit within display 100. This behavior allows a user to open a new window of an application, and interact with the new window, but still have at least a portion of the content within window 704 available to interact with.

[0103] It is noted that gesture 704 may be made on one or more portions of display 114, a configurable area of screen 108, or as shown in FIG 7B (gesture 708A) a gesture capture region 120. The gesture 708 may involve touching portions of screen 108 and or can involve only movement above the surface of portions of screen 108. In some embodiments, for the gesture to be recognized, contact is required to be made with the display 114 for a predetermined period of time.

[0104] In some embodiments, device 100 is configured to interpret different gestures as indications of different operations. For example, the device 100 may be configured to move an image when the gesture 708 is a drag gesture or flick gesture, but launch applications if gesture 708 is a flick, tap gesture, or double tap gesture. In embodiments, when an application is activated, any image displayed in response to the activating of the application (e.g., open window) is placed in focus. This allows a user to immediately begin to interact with the open window of the application. In some embodiments, the location of the gesture 708, whether in gesture capture region 120 or some other portion of display 114, may determine what operation is performed.

[0105] FIGS. 8A-8C illustrates another embodiment of device 100 and its output in response to receiving input indicating a request to launch an application. FIG. 8A illustrates device 100 (with screens 104 and 108) when a window 804 of a first application is displayed on display 114, and a window 808 of a second application is displayed on display 110. As shown in FIG. 8A, in addition to window 804, display 114 also includes a stack 812 of windows underneath window 804.

[0106] FIG. 8B illustrates that a user enters an input, e.g., a gesture 816, such as a tap, drag, or flick gesture on display 110 to indicate a request to activate an application. In some embodiments, activating the application may be effected by two gestures, such as selecting an image, e.g. an icon, and then inputting a gesture in gesture capture region 120.

[0107] As show in FIG. 8B, the gesture 816 is received by touch sensitive display 110, and in response to receiving the gesture 816, an application is launched if it is not already launched, or as an FIGS. 8A-8C a window from a stack, stack 812, is displayed on display 110. As shown in FIG 8C, the activating of the application results in window 820 being taken from the stack 812 and displayed on display 110. The window 820 can then be used by a user to interact with the application. The window 808 which was previously displayed on display 110 is moved to display 114. The window 804 that was previously displayed on display 114 is then moved to stack 812. In some embodiments, the window 808 may be moved to a different stack of display 110 instead of onto display 114. In yet other embodiments, window 808 is placed into stack 812 instead of being displayed on display 114 and window 814 continues to be displayed 804.

[0108] FIGS. 9A-9C illustrates another embodiment of device 100 and its output in response to receiving inputs indicating requests to launch applications. FIG. 9A illustrates device 100 (with

screens 104 and 108) when a first application has been launched. A window 904 of the first application is displayed on display 110. A user can interact with the first application using window 904. In FIG. 9A, no windows are displayed on display 114.

[0109] As shown in FIG. 9B, at some point in time, a user provides an input 908 that indicates a request to open a second application. The input 908 may be for example a gesture received by gesture capture region 120, or one of display 110, display 114, or some other portion of screen 104 and/or screen 108. It should be noted that although input 908 is shown in FIG. 9A as a single input, in other embodiments, more than one input is received in order to indicate a request to launch an application, such as selection of an icon and a gesture indicating a request to open the application represented by the icon.

[0110] Referring now to FIGS. 9 and 10, flow diagrams 900 and 1000 in accordance with at least some embodiments of the present disclosure is shown and will be described. Flows 900 and 1000 are in embodiments performed by a device such as device 100. More specifically, one or more hardware or software components may be involved in performing flows 900 and 1000. In one embodiment, modules in middleware 520 (FIG. 5A) such as multi-display management class 524 (FIG. 5B) perform one or more of the steps of flows 900 and 1000. In other embodiments, in addition to, or in lieu of, middleware 520 performing steps of flows 900 and 1000, operating system kernel(s) 516a, processor 504, and/or drivers 512a-512b may also perform steps of flows 900 and 1000.

[0111] Flow 900 is initiated at 904. Flow 900 passes from 904 to optional step 908 where input regarding launch/minimization behavior is received. In this step, a user may provide configuration information that indicates how a device should handle gestures that indicate a request to launch/activate an application. In embodiments, the device is device 100 (FIGS. 1-3), which includes multiple screens. The user may input any information that affects how a device 100 may handle input gestures, including but not limited to, what applications are launched, how windows are minimized, how Windows are displayed, changes (if any) that occur to displays 104 and 108 when a gesture has been received, what is highlighted when focus is changed, priority list of applications when a gesture is received, whether focus changes are limited to a display, screen, and/or displayed image, whether the type of gesture received results in different operations, where windows of newly launched applications are displayed, and other information.

In one embodiment, the input indicates that in response to a gesture, an application should be launched and the window should be displayed on a specific display.

[0112] From step 908, flow 900 passes to step 912 where input is received indicating a request to activate a multi-screen application. The input may be a gesture that is received in some embodiments by pressure sensitive displays 110 and 114 or gesture capture regions such as region 120. In some embodiments, as shown in FIGS. 1 and 2, the gesture capture region(s) is distinct and located away from the pressure sensitive displays. In other embodiments the gesture capture area may overlap partially or completely with the pressure sensitive displays.

[0113] The gesture may be any type of gesture. In embodiments, the gesture is one or more of the gestures illustrated in FIG. 4A-4H. The device 100 is configured so that the gesture received at step 112 indicates a request to launch an application and change focus to an image (e.g., window) displayed as a result of launching the application.

[0114] Flow 900 passes from step 912 to step 916 where a window of the multi-screen application is displayed on at least a portion of a first screen and at least a portion of a second screen. The application may be newly launched, or previously launched in the displayed window merely redisplayed.

[0115] Following step 916, additional input is received at step 920 to activate a second application. As noted above, the input may be a gesture that is received in some embodiments by pressure sensitive displays 110 and 114 or gesture capture regions such as region 120.

[0116] Flow then passes to step 924, where a window of the second application is displayed on the display that received the input at step 920. For example, if display 110 received the input at step 920, then at step 924 a window of the second application is displayed on display 110. If however, the input received at step 920 is received by display 114, then at step 924 a window of the second application is displayed on display 114.

[0117] As noted above, both displays were previously displaying portions of the window of the multi-screen application. Consequently, when the window of the second application is displayed, the window of the multi-screen application must be modified. As a result, at step 928 a minimized window of the multi-screen application is displayed on the screen not displaying the window of the second application. Using the example above, because the window of the second application is displayed on display 114, then the minimized window of the multi-screen application will be displayed on display 110.

[0118] In some embodiments, the minimized window will display all of the content displayed in the windows displayed at step 916, however it may reduced size so that it fits within a single display. In other embodiments, the minimized window will only display a portion of the content displayed in the windows displayed at step 916, but still allow the user to interact with the multi-screen application. Flow 900 ends at 932.

[0119] Flow 1000 is initiated at 1004. Flow 1000 passes from 1004 to optional step 1008 where input regarding launch/minimization behavior is received. In this step, a user may provide configuration information that indicates how a device should handle gestures that indicate a request to launch/activate an application. In embodiments, the device is device 100 (FIGS. 1-3), which includes multiple screens. The user may input any information that affects how a device 100 may handle input gestures, including but not limited to, what applications are launched, how windows are minimized, how Windows are displayed, changes (if any) that occur to displays 104 and 108 when a gesture has been received, what is highlighted when focus is changed, priority list of applications when a gesture is received, whether focus changes are limited to a display, screen, and/or displayed image, whether the type of gesture received results in different operations, where windows of newly launched applications are displayed, and other information. In one embodiment, the input indicates that in response to a gesture, an application should be launched and the window should be displayed on a specific display.

[0120] From step 1008, flow 1000 passes to step 1012 where input is received indicating a request to activate a first application. The input may be a gesture that is received in some embodiments by pressure sensitive displays 110 and 114 or gesture capture regions such as region 120. In some embodiments, as shown in FIGS. 1 and 2, the gesture capture region(s) is distinct and located away from the pressure sensitive displays. In other embodiments the gesture capture area may overlap partially or completely with the pressure sensitive displays.

[0121] The gesture may be any type of gesture. In embodiments, the gesture is one or more of the gestures illustrated in FIG. 4A-4H. The device 100 is configured so that the gesture received at step 112 indicates a request to launch an application and change focus to an image (e.g., window) displayed as a result of launching the application.

[0122] Flow 1000 passes from step 1012 to step 1016 where a window of the first application is displayed on a first screen. The application may be newly launched, or previously launched, in which case the displayed window is merely a redisplayed window.

[0123] Following step 1016, additional input is received at step 1020 to activate a second application. The input is received by the first screen. Flow then passes to step 1024, where a window of the second application is displayed on the first screen.

[0124] As a result, of displaying the window of the second application of the first screen, the window of the first application is displayed on a second screen at step 1028. In some embodiments, the window of the first application will not be displayed on the second screen but instead will be added to a stack of the first screen. Flow 1000 ends at 1032.

[0125] The exemplary systems and methods of this disclosure have been described in relation to FIGS 1-10. However, to avoid unnecessarily obscuring the present disclosure, the preceding description omits a number of known structures and devices. This omission is not to be construed as a limitation of the scopes of the claims. Specific details are set forth to provide an understanding of the present disclosure. It should however be appreciated that the present disclosure may be practiced in a variety of ways beyond the specific detail set forth herein.

[0126] Furthermore, while the exemplary aspects, embodiments, and/or configurations illustrated herein show the various components of the system collocated, certain components of the system can be located remotely, at distant portions of a distributed network, such as a LAN and/or the Internet, or within a dedicated system. Thus, it should be appreciated, that the components of the system can be combined in to one or more devices, such as a phone, computer, PDA, electronic book reader, gaming device, or collocated on a particular node of a distributed network, such as an analog and/or digital telecommunications network, a packet-switch network, or a circuit-switched network. It will be appreciated from the preceding description, and for reasons of computational efficiency, that the components of the system can be arranged at any location within a distributed network of components without affecting the operation of the system. For example, the various components can be located in a switch such as a PBX and media server, gateway, in one or more communications devices, at one or more users' premises, or some combination thereof. Similarly, one or more functional portions of the system could be distributed between a telecommunications device(s) and an associated computing device.

[0127] Furthermore, it should be appreciated that the various links connecting the elements can be wired or wireless links, or any combination thereof, or any other known or later developed element(s) that is capable of supplying and/or communicating data to and from the connected

elements. These wired or wireless links can also be secure links and may be capable of communicating encrypted information. Transmission media used as links, for example, can be any suitable carrier for electrical signals, including coaxial cables, copper wire and fiber optics, and may take the form of acoustic or light waves, such as those generated during radio-wave and infra-red data communications.

[0128] Also, while the flowcharts have been discussed and illustrated in relation to a particular sequence of events, it should be appreciated that changes, additions, and omissions to this sequence can occur without materially affecting the operation of the disclosed embodiments, configuration, and aspects.

[0129] A number of variations and modifications of the disclosure can be used. It would be possible to provide for some features of the disclosure without providing others.

[0130] For example in one alternative embodiment, all of the focus behavior information and information regarding launching applications may be preprogrammed into the device 100. In these embodiments, there would be no input received from a user regarding how a device 100 handles gestures indicating the launching of applications or a change in focus.

[0131] In other alternative embodiments, device 100 may be in any configuration described above with respect to FIGS. 6A-6J. In these embodiments, gestures would be received as described above indicating a request for launching an application or changing focus.

[0132] In yet another embodiment, the systems and methods of this disclosure can be implemented in conjunction with a special purpose computer, a programmed microprocessor or microcontroller and peripheral integrated circuit element(s), an ASIC or other integrated circuit, a digital signal processor, a hard-wired electronic or logic circuit such as discrete element circuit, a programmable logic device or gate array such as PLD, PLA, FPGA, PAL, special purpose computer, any comparable means, or the like. In general, any device(s) or means capable of implementing the methodology illustrated herein can be used to implement the various aspects of this disclosure. Exemplary hardware that can be used for the disclosed embodiments, configurations and aspects includes computers, handheld devices, telephones (e.g., cellular, Internet enabled, digital, analog, hybrids, and others), and other hardware known in the art. Some of these devices include processors (e.g., a single or multiple microprocessors), memory, nonvolatile storage, input devices, and output devices. Furthermore, alternative software implementations including, but not limited to, distributed processing or component/object

distributed processing, parallel processing, or virtual machine processing can also be constructed to implement the methods described herein.

[0133] In yet another embodiment, the disclosed methods may be readily implemented in conjunction with software using object or object-oriented software development environments that provide portable source code that can be used on a variety of computer or workstation platforms. Alternatively, the disclosed system may be implemented partially or fully in hardware using standard logic circuits or VLSI design. Whether software or hardware is used to implement the systems in accordance with this disclosure is dependent on the speed and/or efficiency requirements of the system, the particular function, and the particular software or hardware systems or microprocessor or microcomputer systems being utilized.

[0134] In yet another embodiment, the disclosed methods may be partially implemented in software that can be stored on a storage medium, executed on programmed general-purpose computer with the cooperation of a controller and memory, a special purpose computer, a microprocessor, or the like. In these instances, the systems and methods of this disclosure can be implemented as program embedded on personal computer such as an applet, JAVA® or CGI script, as a resource residing on a server or computer workstation, as a routine embedded in a dedicated measurement system, system component, or the like. The system can also be implemented by physically incorporating the system and/or method into a software and/or hardware system.

[0135] Although the present disclosure describes components and functions implemented in the aspects, embodiments, and/or configurations with reference to particular standards and protocols, the aspects, embodiments, and/or configurations are not limited to such standards and protocols. Other similar standards and protocols not mentioned herein are in existence and are considered to be included in the present disclosure. Moreover, the standards and protocols mentioned herein and other similar standards and protocols not mentioned herein are periodically superseded by faster or more effective equivalents having essentially the same functions. Such replacement standards and protocols having the same functions are considered equivalents included in the present disclosure.

[0136] The present disclosure, in various aspects, embodiments, and/or configurations, includes components, methods, processes, systems and/or apparatus substantially as depicted and described herein, including various aspects, embodiments, configurations embodiments,

subcombinations, and/or subsets thereof. Those of skill in the art will understand how to make and use the disclosed aspects, embodiments, and/or configurations after understanding the present disclosure. The present disclosure, in various aspects, embodiments, and/or configurations, includes providing devices and processes in the absence of items not depicted and/or described herein or in various aspects, embodiments, and/or configurations hereof, including in the absence of such items as may have been used in previous devices or processes, e.g., for improving performance, achieving ease and\or reducing cost of implementation.

[0137] The foregoing discussion has been presented for purposes of illustration and description. The foregoing is not intended to limit the disclosure to the form or forms disclosed herein. In the foregoing Detailed Description for example, various features of the disclosure are grouped together in one or more aspects, embodiments, and/or configurations for the purpose of streamlining the disclosure. The features of the aspects, embodiments, and/or configurations of the disclosure may be combined in alternate aspects, embodiments, and/or configurations other than those discussed above. This method of disclosure is not to be interpreted as reflecting an intention that the claims require more features than are expressly recited in each claim. Rather, as the following claims reflect, inventive aspects lie in less than all features of a single foregoing disclosed aspect, embodiment, and/or configuration. Thus, the following claims are hereby incorporated into this Detailed Description, with each claim standing on its own as a separate preferred embodiment of the disclosure.

[0138] Moreover, though the description has included description of one or more aspects, embodiments, and/or configurations and certain variations and modifications, other variations, combinations, and modifications are within the scope of the disclosure, e.g., as may be within the skill and knowledge of those in the art, after understanding the present disclosure. It is intended to obtain rights which include alternative aspects, embodiments, and/or configurations to the extent permitted, including alternate, interchangeable and/or equivalent structures, functions, ranges or steps to those claimed, whether or not such alternate, interchangeable and/or equivalent structures, functions, ranges or steps are disclosed herein, and without intending to publicly dedicate any patentable subject matter.

What is claimed is:

1. A method, comprising:

displaying a window of a multi-screen application on at least a portion of a first display of a first screen and at least a second portion of a second display of a second screen;

receiving, by at least one of the first display or the second display, to launch a second application;

in response to receiving the input:

displaying a window of the second application on the at least one of the first display or the second display; and

displaying a minimized window of the multi-screen application on the other of the first display or the second display.

2. The method of claim 1, wherein the second application is a single screen application.

3. The method of claim 1, wherein the second application is a second multi-screen application.

4. The method of claim 1, wherein the minimized window displays only a portion of the content displayed on the window of the multi-screen application.

5. The method of claim 1, further comprising:

placing a focus on the window of the second application.

6. The method of claim 1, wherein the input comprises a tap on the at least one of the first display or the second display.

7. The method of claim 6, wherein the tap is on an icon representing the second application.

8. The method of claim 1, wherein the window of the multi-screen application is added to a stack of the other of the first display or the second display.

9. The method of claim 1, wherein the window of the multi-screen application is displayed in a portrait max configuration.

10. The method of claim 1, wherein the window of the multi-screen application is displayed in a landscape max configuration.

11. A non-transitory computer readable medium storing computer executable instructions that when executed by at least one processor perform a method comprising:

displaying a first window of a first application on a first display of a first screen;

receiving, by a the first display, input to open a second application;

in response to receiving the input:

displaying a second window of the second application on the first display; and

displaying the first window of the first application on a second display.

12. The non-transitory computer readable medium of claim 11, wherein prior to receiving the input, no window is displayed on the second display.

13. The non-transitory computer readable medium of claim 11, wherein prior to receiving the input, a third window is displayed on the second display, and in response to receiving the input, the third window no longer being displayed on the second display.

14. The non-transitory computer readable medium of claim 11, wherein before receiving the input, the second window is opened and positioned within a stack.

15. The non-transitory computer readable medium of claim 11, wherein before receiving the input, the second window is unopened and in response to receiving the input the second application is launched.

16. The non-transitory computer readable medium of claim 14, wherein the stack is for the second display.

17. A dual screen communication device, comprising:

a first touch sensitive display of a first screen;

a second touch sensitive display of a second screen;

a computer readable medium that stores computer executable instructions that when executed by at least one processor perform a method comprising:

displaying a window of a multi-screen application on a first display of a first screen;

receiving, by the first display, input to activate a second application;

in response to receiving the input:

displaying a window of the second application on the first display; and

displaying a minimized window of the multi-screen application on the second display.

18. The device of claim 17, wherein the minimized window displays only a portion of the content displayed on the window of the multi-screen application.

19. The device of claim 17, wherein the minimized window displays the content displayed on the window of the multi-screen application.

20. The device of claim 17, wherein prior to receiving the input, a third window is displayed on the second display, and in response to receiving the input, the third window no longer being displayed on the second display.

ABSTRACT

Embodiments are described for minimizing application windows displayed on one or more displays of a multi-screen device. A device may display a dual screen application which is displayed across both screens. Depending upon which screen is used to launch (or reopen) an application, the dual screen application may be minimized to the other screen.

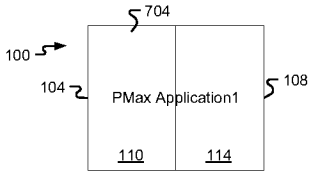


FIG. 7A

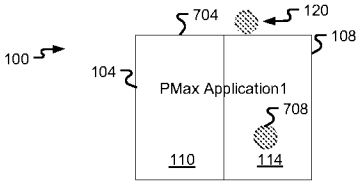


FIG. 7B

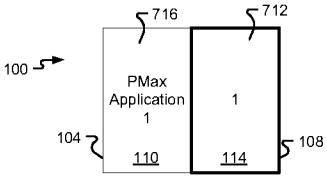


FIG. 7C

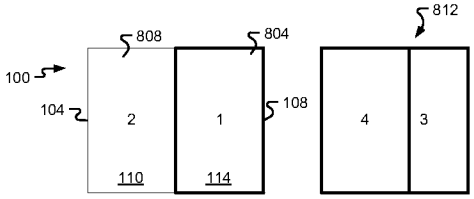


FIG. 8A

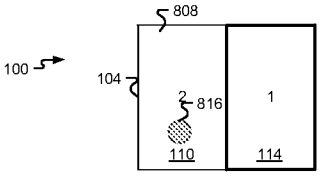


FIG. 8B

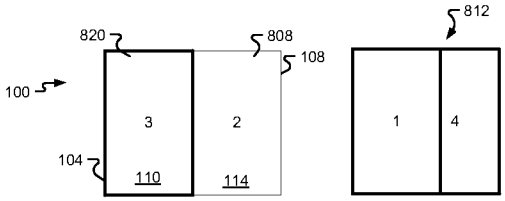
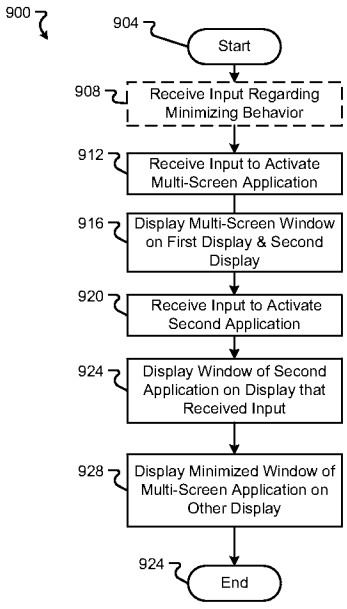
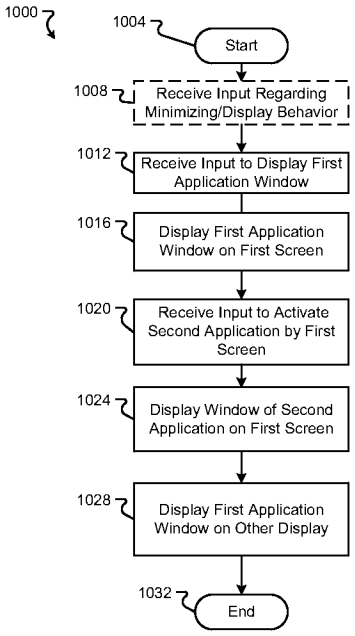


FIG. 8C





Brief Description of the Drawings

[0001] FIG. 10A depicts a first display state of an open device with various non-displayed windows in the form of single applications stacked on either side of the visible display in accordance with embodiments of the present disclosure;

[0002] FIG. 10B depicts a second display state of the open device of Fig. 10A with the single applications in accordance with embodiments of the present disclosure, and further showing a new application launched from the primary screen that results in the desktop reveal function to be cancelled such that the newly launched application shows as being inserted within the window stack and visible on the screen from which the new application was launched, namely the primary screen, and the single application stacked adjacent the secondary screen shows on the secondary screen;

[0003] FIG. 10C depicts another first display state of an open device with various non-displayed windows in the form of single applications stacked on either side of the visible display in accordance with embodiments of the present disclosure;

[0004] FIG. 10D depicts a second display state of the open device of Fig. 10C with the single applications in accordance with embodiments of the present disclosure, and further showing a new application launched from the secondary screen that results in the desktop reveal function to be cancelled such that the newly launched application shows as being inserted within the window stack and visible on the screen from which the new application was launched, namely the secondary screen, and the single application stacked adjacent the primary screen shows on the primary screen;

[0005] FIG. 10E depicts a first display state of an open device with various non-displayed windows in the form of parent and child applications stacked on either side of the visible display in accordance with embodiments of the present disclosure;

[0006] FIG. 10F depicts a second display state of the open device of Fig. 10E with the parent and child applications in accordance with embodiments of the present disclosure, and further showing a merge of the parent and child applications in response to a new application launched from the primary screen that results in the desktop reveal function to be cancelled such that the parent and child applications are shifted to the screen from which the new application was not launched, namely the secondary screen;

[0007] FIG. 10G depicts a third display state of the open device of Fig. 10E with the parent and child applications in accordance with embodiments of the present disclosure, and further showing the newly launched application as being inserted within the window stack and visible on the screen from which the new application was launched, namely the primary screen, and the parent and child applications remaining stacked on the secondary screen with the parent application on top and visible on the secondary screen.

[0008] FIG. 10H depicts a first display state of an open device with various non-displayed windows in the form of parent and child applications stacked on either side of the visible display in accordance with embodiments of the present disclosure;

[0009] FIG. 10I depicts a second display state of the open device of Fig. 10H with the parent and child applications in accordance with embodiments of the present disclosure, and further showing a merge of the parent and child applications in response to a new application launched from the secondary screen that results in the desktop reveal function to be cancelled such that the parent and child applications are shifted to the screen from which the new application was not launched, namely the primary screen;

FIG. 10J depicts a third display state of the open device of Fig. 10H with the parent and child applications in accordance with embodiments of the present disclosure, and further showing the newly launched application as being inserted within the window stack and visible on the screen from which the new application was launched, namely the secondary screen, and the parent and child applications remaining stacked on the primary screen with the parent application on top and visible on the primary screen.

Detailed Description

[0100] Figures 10A-10J depict graphical representations of a device 100 in an open portrait state where the primary screen or display 104 in a first portrait configuration 604 is shown surrounded by a solid line. A secondary screen or display 108 is shown in a second portrait configuration 608, and is also visible as indicated by a solid line.

[0101] Referring first to Figure 10A, the primary screen 104 shows a first page 1004 with a first desktop D1, and the secondary screen 108 shows a first page 1006 with a second desktop D2. Two other pages 1008 and 1010 that are not visible on the primary screen 104 or the secondary screen 108 are also graphically represented as the shaded areas, in which page 1008 is shown to the lefr of page 1004, and page 1010 is shown to the right of page 1006. With respect to pages 1008 and 1010, these pages have been specifically shown as exemplary single page applications.

[0102] Figure 10A further shows the device 100 where a desktop reveal expansion has been initiated. The desktop reveal expansion as noted above may be initiated by, (1) a user input gesture (2) combination of user input gestures (3) a memory input, (4) a response to a predetermined condition (e.g., application control, power level, communications interrupt, operating system state, device screen state open/closed, timers and single or multiple sensor outputs), or (5) any combination thereof such that the MDM class 524 registers and interprets as an initiation command.

[0103] As also noted above, the graphical representations of the desktops and running applications in the pages can conceptually be thought of as one or more combinations of virtual "stacks". These stacks of pages can be created, deleted, shuffled and/or removed by a user or by an output from the MDM class 524 similar to a user manipulating the virtual playing cards of a deck stack.

[0104] The concept of stacks disclosed herein may also be applied to desktops available on the device 100. In other words, a desktop virtual stack may be used for the desktops which can be displayed on the device 100, and an application virtual stack may be used for the applications that are displayed by the device 100. Specifically, when a desktop is revealed, it may be divided into multiple pages for display on multiple screens of the device 100. These pages may be manipulated in their own desktop virtual stack or stacks, separate from the application virtual stacks and/or other virtual stacks. This separation of stacks allows for greater user flexibility in navigating through different applications or desktops by creating an intuitive user interface depending on the data displayed to the screens 104 and 108. However, it is also anticipated that these stacks could be combined to form a single virtual stack. Further, pages in each stack may move together as if the pages in a stack are virtually connected. Therefore, a single position move of a page in one stack correspondingly moves all the pages in that stack by one position.

[0105] In Figs. 10A, 10C, 10E, and 10H, the stacks of individual pages are shown spread outwardly in a linear format on both the left side of the primary screen 104 as well as the right side of the secondary screen 108 to better visualize the organization of the virtual stack example. The position of the page from the primary screen 104 or secondary screen 108 is the position of the page in each respective stack stored on the device 100. Therefore, if the pages are graphically represented immediately adjacent to the primary screen 104 or immediately adjacent to the secondary screen 108, these pages are the first pages in the stack that may be moved onto that adjacent screen with a user input gesture. Although only several pages of the stack are shown in these Figures, the number of pages available to be organized in the stack can vary and, in some embodiments, may equal the number of applications, desktops, and/ or other displayable information created on the device 100. Moreover, the stack adjacent to the primary screen 104 and the stack adjacent to the secondary screen 108 may act as a unified stack, (e.g., when a move or manipulation of a page from one stack moves or manipulates the pages in both stacks), or as independent stacks (each screen having its own independently navigable stack of pages).

[0106] Referring to Figs. 10A and 10B, an embodiment of the present device 100 is shown in the portrait state, where the desktop reveal expansion has been initiated, and the user has decided to launch a new application (3), on page 1012 as shown in Fig. 10B. As indicated in Fig. 10A, the example gesture used to launch the new page 1012 is shown as an exemplary tap or long press gesture 1002. According to embodiments of the present invention, when a user launches a new application or desktop from one of the screens, that newly chosen application or desktop is first displayed on that corresponding screen as a first page or top page in the virtual stack. In Figure 10B, the launching of the new application (3) on the page 1012 is graphically shown as replacing the page 1004 with the previously displayed desktop D1.

[0107] When the new application or desktop is launched, the previous desktop reveal is terminated and therefore the application or desktop that currently resides in the stack closest to the non-launching screen, namely the secondary screen 108, displays that application/desktop. Therefore as also shown in the Fig. 10B, page 1010 running the application (2) is displayed on the secondary screen 108.

[0108] Referring to Figs. 10C and 10D, the launching of an application into the virtual stack is also illustrated, but in Figure 10C, the exemplary user input gesture 1002 is executed on the secondary screen 108. Accordingly, as shown in Fig. 10D, the new page 1012 with the new application (3) replaces the desktop D2 previously displayed on page 1006. Also, the non-launching screen, namely the primary screen 104, then displays page 1008 running the application (1).

[0109] Now referring to Figs. 10E-10G, an additional embodiment is illustrated in which an application is to be launched and inserted into the virtual stack with existing parent and child applications that are running. First, referring to Fig. 10E, pages 1004 and 1006 are running the desktops D1 and D2, and are displayed on screens 104 and 108, respectively. A parent application (1.1) on page 1014 is graphically displayed to the lefr of screen 104, and a child application (1.2) is graphically displayed to the right of the secondary screen 108 on page 1016. A user initiates the user input gesture 1002, such as a tap or long press on the primary screen 104. As graphically represented in Fig. 10F, the parent application (1.1) on page 1014 is moved to "cover" the child application (1.2). Further, the desktop reveal expansion is terminated such that the parent application (1.1) on page 1014 becomes the first page in the stack on the secondary screen 108 and the child application (1.2) on screen 1016 becomes the second or lower stacked application. The new application being launched, namely application (2) on page 1018 replaces the display of desktop D1 on page 1004.

[0110] Now referring to Figs. 10H through 10J, another launch of a new application is illustrated when a parent and child application are running, however the launch occurs from the secondary screen 108. Accordingly, the parent (1.1) and child (1.2) applications become stacked on the primary screen 104 with the parent application on top, while the newly launched application (2) on page 1018 appears on the secondary screen 108.

[0111] Although the Figs. 10A-10J illustrate some sequential movement of the pages, this sequential movement is simply illustrated for purposes of explaining the logic of how the pages are moved to their new locations after execution of the launch. From the user's standpoint, the launching of the new application results in the simultaneous re-ordering of the stack and the display of moved or newly appearing applications or desktops depending upon the user selection for the launch.

[0112] Based on the description of Figs. 10A through 10J, it is evident that the display of windows in a window stack can be selectively changed by simple gesture interactions thereby making multiple desktop and running applications available to the user with minimum effort. Each screen has a logical representation of the windows that have been opened and displayed on the screen. The logical representation is the stack that is made up of one or more windows arranged in an order that can be selected and changed by the user. The user has the ability to launch a new application or desktop in which the most recently launched application or desktop appears on the screen from which the user input gesture was initiated. The other currently running desktops or applications are then logically stacked on the other screen. Thus, the device operates in a very intuitive way for the user to simultaneously manage multiple desktops and applications.

Claims

[0113] 1. A method of displaying a plurality of desktops and/or applications, each having at least one window, on a multi-screen device, comprising:

receiving, by a processor, a first input that represents an instruction to reveal one of a desktop or application on a first display of the multi-screen device;

selecting, by a processor, a first desktop or application to display on the first display;

receiving, by a processor, a second input that represents an instruction to reveal one of a desktop or application on a second display of the multi-screen device;

selecting, by a processor, a second desktop or application to display on the second display;

displaying, by a processor, the selected first and second desktops or applications on the first and second displays;

receiving, by a processor, a third input that represents an instruction to launch yet another desktop or application on either the first or second display of the multi-screen device; and

selecting, by a processor, a third desktop or application to display within a window stack including the selected first and second desktops or applications, and selectively inserting the third desktop or application within the window stack resulting in at least the third desktop or application to be displayed on one of the first and second displays, and one of the first and second desktops or applications to be displayed on the other display.

[0114] 2. The method of claim 1, wherein the first, second, and third inputs comprise at least one of the following: (i) a user input gesture; ii) a combination of user input gestures; iii) a memory output; iv) a response to a programmed condition; and v) hardware timers.

[0115] 3. The method of claim 3, wherein the user input gesture comprises a tap or a long press.

[0116] 4. The method of claim 3, wherein the user input gesture is conducted on off-screen areas of the first and second displays.

[0117] 5. The method of claim 1, wherein the third desktop or application is displayed on the display from which it was launched, and one of the first and second desktops or applications is displayed on the other display.

[0118] 6. The method of claim 6, wherein the desktop or application from the first and second desktops or applications displayed on the other display is the top application in the window stack as identified prior to selecting the third desktop or application.

[0119] 7. The method of claim 6, wherein the first desktop or application is a parent application and the second desktop or application is a child application of the parent application, and the parent application is displayed on the other display.

[0120] 8. A non-transitory computer-readable medium having stored thereon instructions that cause a computing system to execute a method, the instructions comprising:

instructions configured to receive a first input that represents an instruction to reveal a desktop or application on a first display of the multi-screen device;

instructions configured to determine a first desktop or application to display on the first display from the first input resulting in the first desktop or application to be displayed on the first display

instructions configured to receive a second input that represents an instruction to reveal a desktop or application on a second display of the multi-screen device;

instructions configured to determine a second desktop or application to display on a second display of the multi-screen device from the second input resulting in the second desktop or application to be displayed on the second display ;

instructions configured to receive a third input that represents an instruction to launch a new desktop or application in addition to the first and second desktops or applications; and

instructions configured to respond to the third input that causes the new desktop or application to be launched as a third desktop or application, wherein the third desktop or application is displayed on the display from which the third input was executed and said third desktop or application thereby replacing either the first or second desktop or application appearing on that display.

[0121] 9. The computer-readable medium of claim 8, wherein the first display corresponds to a primary screen and wherein the second display corresponds to a secondary screen.

[0122] 10. The computer-readable medium of claim 8, wherein said third desktop or application is inserted into a window stack comprising the first, second, and third desktops or applications

[0123] 11. The computer-readable medium of claim 8, wherein the first, second, and third inputs comprise at least one of the following: i) a user input gesture; ii) a combination of user input gestures; iii) a memory output; iv) a response to a programmed condition; and v) hardware timers.

[0124] 12. The computer-readable medium of claim 8, wherein a lead desktop or application from the first and second desktops or applications is displayed on the display from which the third input was not executed.

[0125] 13. The computer-readable medium of claim 8, wherein the first desktop or application is a parent application and the second desktop or application is a child application of the parent application, and the parent application is displayed on the display from which the third input was not executed.

[0126] 14. A multi-screen user device, comprising:

a first display including a first display area;

a second display including a second display area;

a first user input gesture area of the first display;

a second user input gesture area of the second display, wherein the first and second user input gesture areas are configured to accept input from a user.

a computer-readable medium having instructions stored thereon that include:

a first set of instructions configured to determine a number and identity of desktops or applications selected to be run by a user;

a second set of instructions configured to determine, based on the number and identity of the desktops or applications running, a window stack comprising a logical representation of the desktops and applications within an ordered group whereby a user can selectively move a selected desktop or application within the window stack; and

a third set of instructions responsive to gestures made by the user to launch a new desktop or application wherein the new desktop application is inserted within and added to the window

stack, and the new desktop or application is displayed on the display from which the gestures were made, the new desktop or application thereby replacing either the first or second desktop or application appearing on that display.

[0127] 15. The device of claim 14, wherein the first display corresponds to a primary screen and wherein the second display corresponds to a secondary screen.

[0128] 16. The device of claim 14, wherein said third desktop or application is inserted into a window stack comprising the first, second, and third desktops or applications

[0129] 17. The device of claim 14, wherein the first, second, and third inputs comprise at least one of the following: i) a user input gesture; ii) a combination of user input gestures; iii) a memory output; (iv) a response to a programmed condition; and v) hardware timers.

[0130] 18. The device of claim 14, wherein a lead desktop or application from the first and second desktops or applications is displayed on the display from which the gestures were not made.

[0131] 19. The device of claim 14, wherein the first desktop or application is a parent application and the second desktop or application is a child application of the parent application, and the parent application is displayed on the display from which the gestures were not made.

[0132] 20. The device of claim 14, wherein the displayed desktops or applications can be further changed by execution of another user gesture in which at least one of the first, second, or third desktops or applications becomes non-visible.

ABSTRACT

A dual-screen user device and methods for revealing a combination of desktops on single and multiple screens are disclosed. Specifically, a determined number of desktops and/or running applications can be selectively displayed on dual screen displays conditioned upon inputs received and the state of the device. Desktop displays and applications can be selectively shifted between the screens by user gestures, or moved off of the screens by other user gestures and hidden. The hidden desktops and screens can be re-displayed by yet another gesture. The desktops and applications are arranged in a window stack that represents a logical order of the desktops and applications. Desktops and applications can be selectively launched and added to the window stack. The user can also select where the desktops/applications are to be inserted and where they are first to be displayed afrer being launched.

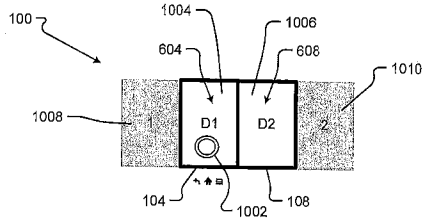


FIG. 10A

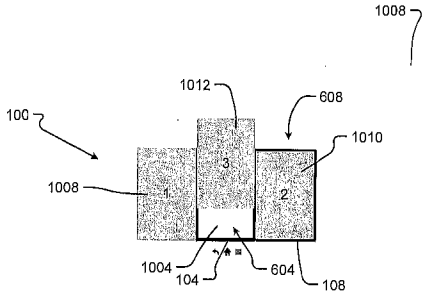


FIG. 10B

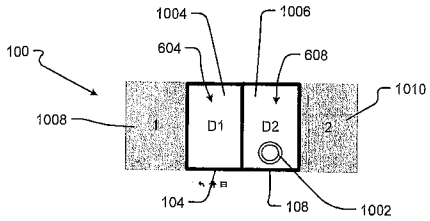


FIG. 10C

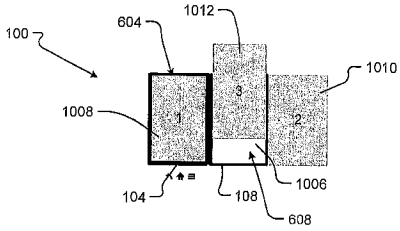


FIG. 10D

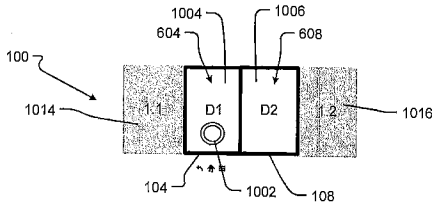


FIG. 10E

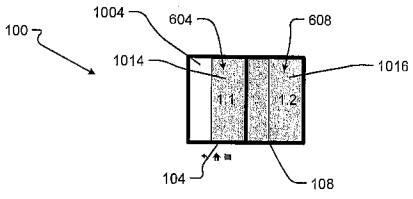


FIG. 10F

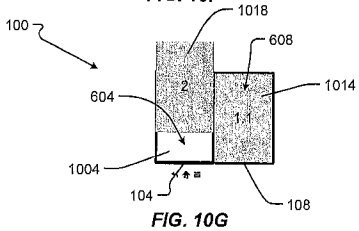


FIG. 10G

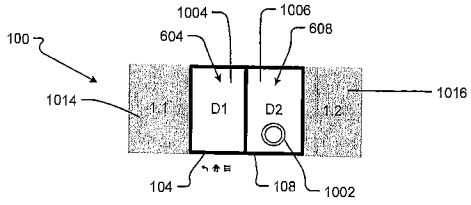


FIG. 10H

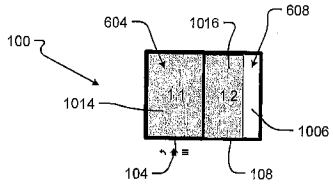


FIG. 10I

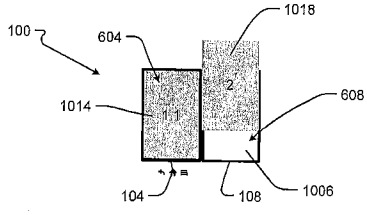


FIG. 10J

Brief Description of the Drawings

[0001] FIG. 10A depicts a first display state of an open device with various non-displayed window stacked on one side of the visible display in accordance with embodiments of the present disclosure;

[0002] FIG. 10B depicts a second display state of an open device with various non-displayed window stacked on the one side of the visible display in accordance with embodiments of the present disclosure, and wherein a user executes an input gesture to re-launch a selected window;

[0003] FIG. 10C depicts a third display state of an open device with various non-displayed window stacked on the one side of the visible display, wherein the selected window re-launched is displayed on the screen from which the gesture was executed, thereby changing an order of the window stack;

[0004] Figs. 10A-C depict graphical representations of embodiments of a device 100 in an open portrait state where the primary screen or display 104 in a first portrait configuration 604 is shown surrounded by a solid line, while the secondary screen or display 108 is in a second portrait configuration 608 and is also visible as indicated by a solid line.

Detailed Description

[0005] Referring first to Fig. 10A, the primary screen 104 shows a first page 1004 showing a first desktop (D1), and the secondary display 108 shows a first page 1006 of a single page application (1). Two other pages 1008 and 1010 that are not visible on the primary screen 104 or the secondary screen 108 are also graphically represented and are shown adjacent the secondary display 108. Fig.10A further shows the present device 100 where a desktop reveal expansion has been initiated. The desktop reveal expansion may be initiated by, (1) a user input gesture, (2) combination of user input gestures, (3) a memory output, (4) a response to a predetermined condition (e.g., application control, power levels, communications interrupt, operating system state, device screen state open/closed, timers, and single or multiple sensor outputs) or (5) any combination thereof, such that the MDM class 524 registers and interprets as an initiation command. As noted above, the graphical representations of the desktops and running applications in the pages can conceptually be thought of as one or more combinations of virtual "stacks" comprising separate viewable windows that run respective desktops or applications. These stacks of pages can be created, deleted, shuffled, and/or moved by a user or by an output from the MDM class 524, similar to a user manipulating the virtual playing cards of a deck stack.

[0006] The concept of stacks disclosed herein may also be applied to desktops available on the device 100. In other words, a desktop virtual stack may be used for the desktops which can be displayed by the device 100 and an application virtual stack may be used for the applications which can be displayed by the device 100. Specifically, when a desktop is revealed it may be divided into multiple pages for display on multiple screens of the device 100. These pages may be manipulated in their own desktop virtual stack, or stacks, separate from the application virtual stacks and/or other virtual stacks. This separation of stacks allows for greater user flexibility in navigating through different applications or desktops by creating an intuitive interface dependent on the data displayed to the screens 104, 108. However, it is also anticipated that these stacks could be combined to form a single virtual stack. Further, pages in each stack may move together as if the pages in a stack are virtually connected. Therefore, a single position move of a page in one stack correspondingly moves all of the pages in that stack by one position.

[0007] In Figs. 10A-C, the stacks of individual pages 1004, 1006, 1008, and 1010 are shown spread outwardly in a linear format with the non –visible applications (2) and (3) of screens 1008 and 1010 located to the right side of the secondary screen 108 to better visualize the organization of the virtual window stack. Although the non-visible applications are shown adjacent only the secondary screen, the running applications and desktops could be shown adjacent the primary screen 104, or adjacent both of the screens thereby indicating the presence of one or more window stacks. Thus, the position of a page from the primary screen 104 or secondary screen 108 is the position of the page in each respective stack stored on the device 100, and if a page is graphically represented immediately adjacent to the primary screen 104, or immediately adjacent to the secondary screen 108, it is the first page in the stack that may be moved onto that adjacent screen with a user input gesture. Although only several pages of a stack are shown in Figs. 10A-C, the number of pages available to be organized in the stack can vary and, in some embodiments, may equal the number of applications, desktops, and/or other displayable information created on the device 100. Moreover, the stack adjacent to the primary screen 104 and the stack adjacent to the secondary screen 108 may act as a unified stack (where a move or manipulation of a page from one stack moves or manipulates the pages in both stacks), or as independent stacks (each screen having its own independently navigable stack of pages).

[0008] Fig. 10A specifically shows an embodiment of the present device 100, in an open portrait state, where the first desk top (D1) is displayed on page 1004, the first application (1) on page 1006, the second application (2) on page 1008, and the third application (3) on page 1010. Fig. 10A also illustrates that a user has initiated a user input gesture 1002 on the primary screen 104 in order to re-position the application (2) on the page 1008. More specifically, a user has initiated a re-launch of the second application, and referring to Fig. 10B, this re-launching operation is graphically represented by the second application (2) on page 1008 having a reduced size as compared to the other running desktops and applications.

[0009] After the user has executed the re-launch of the selected desktop or application, the selected desktop or application then becomes the first page shown on the screen from which the application was re-launched. Accordingly, as shown in Fig. 10C, the second application (2) on the page 1008 is shown as the first page occupying the primary screen 104. The page 1004 running the desk top (D1) is "covered" or in other words, is located as an underlying page in the window stack. This covering of the page 1004 could also be graphically represented by displaying the page 1004 to the left of the primary screen 104. Therefore, it is apparent that a user may easily manipulate the order of a window stack by re-launching selected applications or desktops.

[0010] Although the Figures illustrate the re-launch operation as being executed upon the primary screen 104, it should also be understood that a user could also re-launch selected desktops or applications from the secondary screen 108 in which those selected applications or desktops would then appear as the top page on the secondary screen. Further, a user may re-launch one or more applications or desktops a number of times in order to further rearrange the specific order of the window stack according to the needs of the user.

[0011] Although Figs. 10A-10C are presented in the open portrait state, it shall be understood that the features disclosed in these figures can also be incorporated in other modes of operation of the device 100 to include a landscape state or any of the other "physical" states disclosed herein.

[0012] Based on the description of Figs. 10A-10C, it is evident that the display of windows in a window stack can be selectively changed by simple gesture interactions thereby making multiple desktops and running applications available to the user with minimum effort. Each screen has a logical representation of the windows that have been opened and displayed on the screens. The logical representation is the stack that is made up of one or more windows arranged in an order that can be selected and changed by the user. The windows can be indexed or shifted to display desire windows, as well as to move windows between separate stacks. The ordered arrangement of a stack maintains the logical construction of the stack intact when the user navigates the window stack, for example in an off-screen area. As shown specifically in the Figs. 10B and 10C, a user may easily manipulate the order of the window stack(s) by re-launching a selected desktop or application. Thus, the device operates in a very intuitive way for the user to simultaneously manage multiple desktops and applications.

Claims

[0013] 1. A method of displaying a plurality of desktops and/or applications, each having at least one window, on a multi-screen device, comprising:

receiving, by a processor, a first input that represents an instruction to reveal one of a desktop or application on a first display of the multi-screen device;

selecting ,by a processor, a first desktop or application to display on the first display;

receiving a second input that represents an instruction to reveal one of a desktop or application on a second display of the multi-screen device;

selecting ,by a processor,a second desktop or application to display on the second display;

displaying ,by a processor,the selected first and second desktops or applications on the first and second displays;

receiving ,by a processor,a third input that represents an instruction to launch one of a desktop or application;

selecting ,by a processor,a third desktop or application to add to a window stack comprising the first, second, and third desktops or applications;

initially ,by a processor, positioning the third desktop or application in the window stack in a non-displayed position

selectively ,by a processor, moving the third desktop or application to another position by re-launching the third desktop or application from the first or second displays, resulting in the third desktop or application being re-displayed on the display from which it was re-launched, thereby changing an order of the desktops and applications running in the window stack.

[0014] 2. The method of claim 1, wherein the first, second, and third inputs comprise at least one of the following: i) a user input gesture; ii) a combination of user input gestures; iii) a memory output; iv) a response to a programmed condition; and v) hardware timers.

[0015] 3. The method of claim 1, wherein at least one of the first, second, and third inputs is achieved by at least a user input gesture.

[0016] 4. The method of claim 1, wherein the third input is a user gesture comprising at least one of a tap or long hold gesture.

[0017] 5. The method of claim 1, wherein the window stack comprises a number of other desktops or applications that have been launched by respective inputs from the user, and wherein the selectively moving step changes the order of the desktops and applications running in the window stack including the number of other desktops or applications.

[0018] 6. The method of claim 1, wherein the user further manipulates the order of the window stack by at least another user input gesture in which at least one of the desktops or applications is selected and re-launched, resulting in that that selected desktop or application being re-displayed on the display from which it was re-launched.

[0019] 7. A non-transitory computer-readable medium having stored thereon instructions that cause a computing system to execute a method, the instructions comprising: instructions configured to receive a first predetermined input that represents an instruction to determine and reveal a first desktop or application on a first display of the multi-screen device;

instructions configured to receive a second predetermined input that represents an instruction to determine and reveal a second desktop or application on a second display of the multi-screen device;

instructions configured to respond to the first and second predetermined inputs with outputs that cause the first desktop or application to be displayed on the first display and cause the second desktop or application to be displayed on the second display

instructions configured to receive a third predetermined input that represents an instruction to determine and launch a third desktop or application;

instructions configured to respond to the third predetermined input that causes the third desktop or application to be placed in a non-visible position within a window stack comprising the first, second, and third desktops or applications;

instructions configured to receive a fourth predetermined input that represents an instruction to re-launch the third desktop or application;

instructions configured to respond to the fourth predetermined input that causes the third desktop or application to be placed in a visible position on the display from which the fourth predetermined input was executed.

[0020] 8. The computer-readable medium of claim 7, wherein the first display corresponds to a primary screenand wherein the second display corresponds to a secondary screen.

[0021] 9. The computer-readable medium of claim 7, wherein the window stack comprises a number of other desktops or applications that have been launched by respective instructions, and wherein the instructions configured to respond to the fourth predetermined input also causes a change in the order of the desktops and applications running in the window stack including the number of other desktops or applications.

[0022] 10. The computer-readable medium of claim 7, wherein the order of the window stack is further manipulated by additional instructions configured to receive yet additional inputs that represent instructions to re-launch additional desktops or applications, and instructions configured to respond to the additional inputs casuing the additional desktops and applications to be placed in selected visible positions on the displays from which the additional instructions were executed.

[0023] 11. The computer-readable medium of claim 7, wherein:

at least one of the predetermined inputs comprise at least a user input gesture.

[0024] 12. A multi-screen user device, comprising:

a first display including a first display area;

a second display including a second display area;

a first user input gesture area of the first display;

a second user input gesture area of the second display, wherein the first and second user input gesture areas are configured to accept input from a user.

a computer-readable medium having instructions stored thereon that include:

a first set of instructions configured to determine a number and identity of desktops or applications selected to be run by a user;

a second set of instructions configured to determine, based on the number and identity of the desktops or applications running, a window stack comprising a logical representation of the desktops and applications within an ordered group whereby a user can selectively move a selected desktop or application to either a visible position on another display or a non-visible position; and

a third set of instructions responsive to a gesture made by the user to move a selected desktop or application from an original position in the window stack to another position in the window stack, wherein the third set of instructions include an instruction to re-launch the selected desktop or application causing the selected desktop or application to be displayed on the display from which the selected desktop or application was launched.

[0025] 13. The device of claim 12, wherein the first display corresponds to a primary screen and wherein the second display corresponds to a secondary screen.

[0026] 14. The device of claim 12, wherein the first user input gesture area is located on an off-screen area of the first display and outside of the first display area.

[0027] 15. The device of claim 12, wherein the second user input gesture area is located on an off-screen area of the second display and outside of the second display area.

[0028] 16. The device of claim 12, wherein the first and second displays show different displayed images by user input gestures that manipulate an order of the desktops and applications in the window stack.

[0029] 17. The device of claim 12, wherein the window stack comprises a number of other desktops or applications that have been launched by additional respective instructions, including at least one instruction to re-launch another selected desktop or application causing the additional selected desktop or application to be displayed on the display from which the additional selected desktop or application was launched.

ABSTRACT

A dual-screen user device and methods for revealing a combination of desktops on single and multiple screens are disclosed. A number of desktops and/or running applications can be selectively displayed on dual screen displays conditioned upon inputs received and the state of the device Desktop displays and applications can be selectively shifted between the screens by user gestures or moved off of the screens by other user gestures and therefore hidden. The hidden desktops and screens however can be re-displayed by yet another gesture. The user gestures prevent the user from having to open and close the running desktops and applications, or to execute other user commands that otherwise result in a laborious effort by the user to manage the multiple desktops and applications. One user gesture or input enables a user to change an order of the window stack by simply re-launching a selected desktop or application.

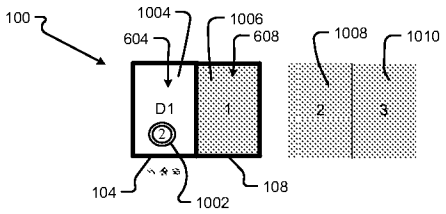


FIG. 10A

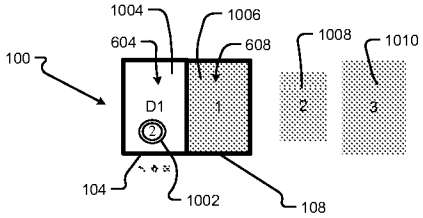


FIG. 10B

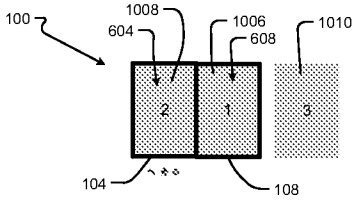


FIG. 10C

[0100] FIGS. 7A-7C illustrate an embodiment showing device 100 and how device 100 responds when a modal window is launched. FIG. 7A illustrates device 100 when a multi-screen application has been launched and a window 704 of the application is displayed on the display 110 of screen 104 and the display 114 of screen 108. In FIG. 7A, device 100 is operating in a portrait max mode, meaning that a window is displayed across both displays 100 and 114. There may be other windows in the stack underneath window 704.

[0101] As shown in FIG. 7B, a user can make a gesture 708, such as a tap, drag, or flick gesture on display 114 that results in launching of a modal window. In some embodiments, the gesture may be made in gesture capture region 120 (e.g., gesture 708A). Device 100 receives the gesture 708 or 708A. In response to receiving the gesture, device 100 will launch the modal window 712. As shown in FIG 7C, the modal window 712 is displayed on display 114. The window 712 will provide a user interface that a user can interact with.

[0102] As shown in FIG. 7C, the window 712 is displayed on display 114 because the input, i.e. gesture 708 was received on display 114. Window 712 is shown as displayed on only a portion of display 114. In the embodiment shown in FIG. 7C, the portions of displays 110 and 114 that do not display the modal window 712 are inactivated. That is, no input is received from those portions of the display. As indicated above, the modal window is intended to focus a user on the content of the modal window. In the embodiment shown in FIG. 7C, the portions of the displays (110 and 114) that do not display modal window 712, are dimmed. The dimming 716 is performed in order to provide feedback to the user indicating that those portions are inactive.

[0103] It is noted that the input that results in launching of the modal window 712 can be any type of input. As noted above, the input may be a gesture 708 from a user. The gesture may be a gesture that expands a menu, makes a selection from a list, or indicates a request to perform some other operation. In other embodiments, the input may cause an error that results in the display of modal window 712. For example, a user may attempt to perform an operation that is not allowed. The modal window 712 is displayed to inform the user and give the user options. As can be appreciated, the modal window 712 may be displayed by an application running on device 100, or in some embodiments by an operating system that monitors and controls overall system performance device 100.

[0104] FIGS. 8A-8C illustrates another embodiment of device 100 and its output in response to receiving input that results in launching a modal window. FIGS. 8A and 8B are similar to FIGS. 7A and 7B. However, as shown in FIG. 8C, window 712 is displayed and occupies all of display 114. As a result, only display 110 is dimmed to indicate that it is inactive.

[0105] FIGS. 9A-9C illustrates another embodiment of device 100 and its output in response to receiving input that results in launching a modal window. FIGS. 9A and 9B are similar to FIGS. 7A and 7B. However, as shown in FIG. 8C, window 712 is displayed and occupies all of display 114. Furthermore, instead of display 110 being indicated as inactive, by for example being dimmed, the multi-screen application window 704 has been minimized and is in the stack of display 114. In other words, when modal window 712 is closed, a minimized window of the multi-screen application will be displayed on display 114. This embodiment allows a user to interact with display 110 to for example open up other applications. In other words, having modal window 712 displayed (on display 114) does not lock up the entire device, and the user may utilize other applications using screen 104 and display 110.

[0106] FIGS. 10A-10C illustrates another embodiment of device 100 and its output in response to receiving input that results in launching a modal window. FIG. 10A illustrates device 100 with at least two applications open and windows for the applications displayed on displays 110 and 114. Window 804 is displayed on display 114, while window 808 is displayed on display 110. FIG. 10B illustrates that a user has entered input on display 110, in the form of gesture 816. As a result of the input, a modal window 820 is displayed (FIG. 10C) on display 110.

[0107] Because the window 804 displayed on display 114 was not involved in receiving the input that resulted in display of modal window 820, it is not affected by the display of modal window 820. In contrast, portions of display 110 that are not displaying modal window 820 are made inactive. Also, they are dimmed 824 to let the user know that they are inactive. In this embodiment, a user may continue to interact with window 804 and its associated application but cannot interact with window 808 and its associated application until modal window 820 is closed.

[0108] Referring now to FIG. 11, flow diagram 1000 in accordance with at least some embodiments of the present disclosure is shown and will be described. Flow 1000 is in embodiments performed by a device such as device 100. More specifically, one or more hardware or software components may be involved in performing flow 1000. In one

embodiment, modules in middleware 520 (FIG. 5A) such as multi-display management class 524 (FIG. 5B) perform one or more of the steps of flow 1000. In other embodiments, in addition to, or in lieu of, middleware 520 performing steps of flow 1000, operating system kernel(s) 516a, processor 504, and/or drivers 512a-512b may also perform steps of flow 1000.

[0109] Flow 1000 is initiated at 1004. Flow 1000 passes from 1004 to optional step 1008 where input regarding modal window behavior is received. In this step, a user may provide configuration information that indicates how a device should handle display of modal windows on one or more displays. In embodiments, the device is device 100 (FIGS. 1-3), which includes multiple screens. The user may input any information that affects how a device 100 may handle modal window launching, modal window displaying, including but not limited to, how applications are minimized, changes (if any) that occur to displays 110 and 114 when a modal window is displayed, and other information.

[0110] From step 1008, flow 1000 passes to step 1012 where input is received to launch a multi-screen application. The input may be for example a gesture on a display or on a gesture capture region. In other embodiments, the input may be received by buttons of a screen. Step 1012 is followed by step 1014, where a window of multi-screen application is displayed. In embodiments, the window is displayed on two displays. That is, the window is displayed on at least a portion of a first display and a portion of a second display. In other embodiments, the window may be displayed on a single display, which in embodiments is a minimized version of a larger window that is displayed on two displays (the first display and the second display).

[0111] Flow 1000 passes to optional step 1016 where input is received to launch a second application. In response, to the input received at step 1016, flow passes to optional step 1020 where a window of the second application is displayed. The application may be a multi-screen application or an application that occupies only one screen.

[0112] At step 1024, input is received that results in launching of modal window. The input may be a gesture from a user. The gesture may be a gesture that expands a menu, makes a selection from a list, or indicates a request to perform some other operation. In other embodiments, the input may cause an error that results in the display of modal window. For example, a user may attempt to perform an operation that is not allowed. The modal window is displayed to inform the user and give the user options. As can be appreciated, the modal window

may be displayed by an application running on device 100, or in some embodiments by an operating system that monitors and controls overall system performance device 100.

[0113] Flow 1000 passes to step 1026 where the modal window is displayed. The modal window may be displayed on a portion of a display or on an entire display. In other embodiments, the modal window may be displayed on portions of more than one display.

[0114] Flow 1000 includes a number of optional steps that may be performed in some embodiments and not others. Optional step 1028 involves minimizing a window of a multi-screen application such as the one displayed at step 1014. Step 1028 involves in some embodiments, changing the window of the multi-screen application to display on a single screen instead of multiple screens. In these embodiments, this may allow one of the screens to remain active in response to the modal window being displayed.

[0115] Flow 1000 passes to optional step 1032, where displays are made inactive. In some embodiments, only portions of the displays that are not displaying the modal window are made inactive. In other embodiments, an entire display may be made inactive.

[0116] Following step 1032, optional step 1034 dims portions of the display. The dimming of the portions of the display provides a visual queue to the user that the portions that are dimmed are inactive. In other embodiment, instead of dimming, optional step 1034 may involve somehow distinguishing the display portions that are inactive.

[0117] Flow 1000 then includes step 1036 where input is received by the modal window. The input may be to close the window, or other input that satisfies a request made in the modal window. Step 1036 is followed by step 1040 where the modal window is closed. Flow 1000 ends at 1044.

[0118] The exemplary systems and methods of this disclosure have been described in relation to FIGS 1-10. However, to avoid unnecessarily obscuring the present disclosure, the preceding description omits a number of known structures and devices. This omission is not to be construed as a limitation of the scopes of the claims. Specific details are set forth to provide an understanding of the present disclosure. It should however be appreciated that the present disclosure may be practiced in a variety of ways beyond the specific detail set forth herein.

[0119] Furthermore, while the exemplary aspects, embodiments, and/or configurations illustrated herein show the various components of the system collocated, certain components of the system can be located remotely, at distant portions of a distributed network, such as a LAN

and/or the Internet, or within a dedicated system. Thus, it should be appreciated, that the components of the system can be combined in to one or more devices, such as a phone, computer, PDA, electronic book reader, gaming device, or collocated on a particular node of a distributed network, such as an analog and/or digital telecommunications network, a packet-switch network, or a circuit-switched network. It will be appreciated from the preceding description, and for reasons of computational efficiency, that the components of the system can be arranged at any location within a distributed network of components without affecting the operation of the system. For example, the various components can be located in a switch such as a PBX and media server, gateway, in one or more communications devices, at one or more users' premises, or some combination thereof. Similarly, one or more functional portions of the system could be distributed between a telecommunications device(s) and an associated computing device.

[0120] Furthermore, it should be appreciated that the various links connecting the elements can be wired or wireless links, or any combination thereof, or any other known or later developed element(s) that is capable of supplying and/or communicating data to and from the connected elements. These wired or wireless links can also be secure links and may be capable of communicating encrypted information. Transmission media used as links, for example, can be any suitable carrier for electrical signals, including coaxial cables, copper wire and fiber optics, and may take the form of acoustic or light waves, such as those generated during radio-wave and infra-red data communications.

[0121] Also, while the flowcharts have been discussed and illustrated in relation to a particular sequence of events, it should be appreciated that changes, additions, and omissions to this sequence can occur without materially affecting the operation of the disclosed embodiments, configuration, and aspects.

[0122] A number of variations and modifications of the disclosure can be used. It would be possible to provide for some features of the disclosure without providing others.

[0123] For example in one alternative embodiment, all of the phone call receiving behavior information and information regarding launching applications may be preprogrammed into the device 100. In these embodiments, there would be no input received from a user regarding how a device 100 handles gestures indicating the launching of applications or a change in focus.

[0124] In other alternative embodiments, device 100 may be in any configuration described above with respect to FIGS. 6A-6J. In these embodiments, windows of open applications may be minimized, displayed, or modes changed in any of the described configurations.

[0125] In yet another embodiment, the systems and methods of this disclosure can be implemented in conjunction with a special purpose computer, a programmed microprocessor or microcontroller and peripheral integrated circuit element(s), an ASIC or other integrated circuit, a digital signal processor, a hard-wired electronic or logic circuit such as discrete element circuit, a programmable logic device or gate array such as PLD, PLA, FPGA, PAL, special purpose computer, any comparable means, or the like. In general, any device(s) or means capable of implementing the methodology illustrated herein can be used to implement the various aspects of this disclosure. Exemplary hardware that can be used for the disclosed embodiments, configurations and aspects includes computers, handheld devices, telephones (e.g., cellular, Internet enabled, digital, analog, hybrids, and others), and other hardware known in the art. Some of these devices include processors (e.g., a single or multiple microprocessors), memory, nonvolatile storage, input devices, and output devices. Furthermore, alternative software implementations including, but not limited to, distributed processing or component/object distributed processing, parallel processing, or virtual machine processing can also be constructed to implement the methods described herein.

[0126] In yet another embodiment, the disclosed methods may be readily implemented in conjunction with software using object or object-oriented software development environments that provide portable source code that can be used on a variety of computer or workstation platforms. Alternatively, the disclosed system may be implemented partially or fully in hardware using standard logic circuits or VLSI design. Whether software or hardware is used to implement the systems in accordance with this disclosure is dependent on the speed and/or efficiency requirements of the system, the particular function, and the particular software or hardware systems or microprocessor or microcomputer systems being utilized.

[0127] In yet another embodiment, the disclosed methods may be partially implemented in software that can be stored on a storage medium, executed on programmed general-purpose computer with the cooperation of a controller and memory, a special purpose computer, a microprocessor, or the like. In these instances, the systems and methods of this disclosure can be implemented as program embedded on personal computer such as an applet, JAVA® or CGI

script, as a resource residing on a server or computer workstation, as a routine embedded in a dedicated measurement system, system component, or the like. The system can also be implemented by physically incorporating the system and/or method into a software and/or hardware system.

[0128] Although the present disclosure describes components and functions implemented in the aspects, embodiments, and/or configurations with reference to particular standards and protocols, the aspects, embodiments, and/or configurations are not limited to such standards and protocols. Other similar standards and protocols not mentioned herein are in existence and are considered to be included in the present disclosure. Moreover, the standards and protocols mentioned herein and other similar standards and protocols not mentioned herein are periodically superseded by faster or more effective equivalents having essentially the same functions. Such replacement standards and protocols having the same functions are considered equivalents included in the present disclosure.

[0129] The present disclosure, in various aspects, embodiments, and/or configurations, includes components, methods, processes, systems and/or apparatus substantially as depicted and described herein, including various aspects, embodiments, configurations embodiments, subcombinations, and/or subsets thereof. Those of skill in the art will understand how to make and use the disclosed aspects, embodiments, and/or configurations afrer understanding the present disclosure. The present disclosure, in various aspects, embodiments, and/or configurations, includes providing devices and processes in the absence of items not depicted and/or described herein or in various aspects, embodiments, and/or configurations hereof, including in the absence of such items as may have been used in previous devices or processes, e.g., for improving performance, achieving ease and\or reducing cost of implementation.

[0130] The foregoing discussion has been presented for purposes of illustration and description. The foregoing is not intended to limit the disclosure to the form or forms disclosed herein. In the foregoing Detailed Description for example, various features of the disclosure are grouped together in one or more aspects, embodiments, and/or configurations for the purpose of streamlining the disclosure. The features of the aspects, embodiments, and/or configurations of the disclosure may be combined in alternate aspects, embodiments, and/or configurations other than those discussed above. This method of disclosure is not to be interpreted as reflecting an intention that the claims require more features than are expressly recited in each claim. Rather,

as the following claims reflect, inventive aspects lie in less than all features of a single foregoing disclosed aspect, embodiment, and/or configuration. Thus, the following claims are hereby incorporated into this Detailed Description, with each claim standing on its own as a separate preferred embodiment of the disclosure.

[0131] Moreover, though the description has included description of one or more aspects, embodiments, and/or configurations and certain variations and modifications, other variations, combinations, and modifications are within the scope of the disclosure, e.g., as may be within the skill and knowledge of those in the art, afrer understanding the present disclosure. It is intended to obtain rights which include alternative aspects, embodiments, and/or configurations to the extent permitted, including alternate, interchangeable and/or equivalent structures, functions, ranges or steps to those claimed, whether or not such alternate, interchangeable and/or equivalent structures, functions, ranges or steps are disclosed herein, and without intending to publicly dedicate any patentable subject matter.

What is claimed is:

1. A method, comprising:

displaying a window of a multi-screen application on at least one of a first display of a first screen and a second display of a second screen;

receiving input that result in launching a modal window; and

in response to the input, displaying, on at least one of the first display or the second display, a modal window.

2. The method of claim 1, wherein the window of the multi-screen application is displayed on at least a portion of the first display and at least a second portion of the second display.

3. The method of claim 2, wherein the input is received by the first screen and the modal window is displayed on the first display.

4. The method of claim 3, further comprising:

in response to displaying the modal window, dimming at least a portion of the second display.

5. The method of claim 4, further comprising:

in response to displaying the modal window, dimming at least a portion of the first display.

6. The method of claim 1, wherein the window of the multi-screen application is minimized to be displayed on the first display.

7. The method of claim 6, further comprising:

displaying a second window of a second application on the second display.

8. The method of claim 7, wherein the input is received by the first screen and the modal window is displayed on the first display.

9. The method of claim 8, further comprising:

in response to displaying the modal window, dimming at least a portion of the first display.

10. The method of claim 9, wherein the second display is not dimmed.

11. The method of claim 9, wherein the modal window occupies only a portion of the one of the first screen or the second screen.

12. The method of claim 1, wherein the modal window is launched by an operating system.

13. A non-transitory computer readable medium storing computer executable instructions that when executed by at least one processor perform a method comprising:

displaying a window of a multi-screen application on at least a first portion of a first display and on at least a second portion of a second display;

receiving input that results in launching of a modal window; and

in response to the input, displaying, on at least one of the first screen or the second screen, a modal window.

14. The non-transitory computer readable medium of claim 11, wherein the modal window is displayed on at least the first portion of the first display and on at least the second portion of the second display.

15. The non-transitory computer readable medium of claim 11, further comprising:

receiving input by the modal window.

16. The non-transitory computer readable medium of claim 15, further comprising:

after receiving the input, closing the modal window.

17. The non-transitory computer readable medium of claim 11, further comprising

in response to displaying the modal window, deactivating portions of the first display and the second display not displaying the modal window.

18. A dual screen communication device, comprising:

a first touch sensitive display of a first screen;

a second touch sensitive display of a second screen;

a computer readable medium that stores computer executable instructions that when executed by at least one processor perform a method comprising:

receiving input that result in launching a modal window; and

in response to the input, displaying, on at least one of the first touch sensitive display or the second touch sensitive display, the modal window; and

in response to displaying the modal window, deactivating portions of the first touch sensitive display and the second touch sensitive display not displaying the modal window.

19. The device of claim 18, wherein the modal window is launched by an operating system.

20. The device of claim 18, wherein the modal window is launched by an application.

ABSTRACT

Embodiments are described for handling display of modal windows in a multi-screen device. In embodiments, a modal window will be launched and displayed in a display which receives the input that resulted in the display of the modal window. The other portions of a first display or second display, not displaying the modal window, are made inactive. In other embodiments, the modal window occupies only a first display and the second display remains active.

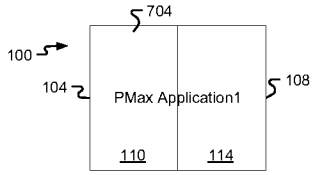


FIG. 7A

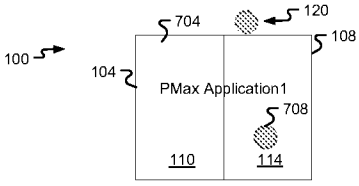


FIG. 7B

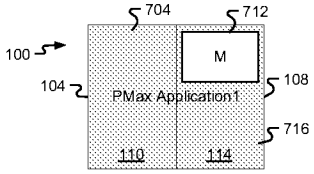


FIG. 7C

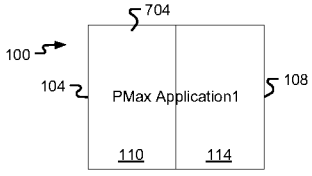


FIG. 8A

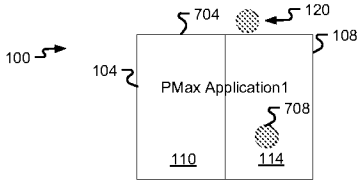


FIG. 8B

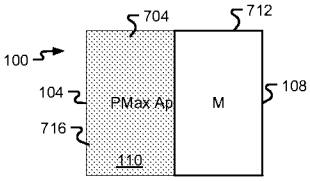


FIG. 8C

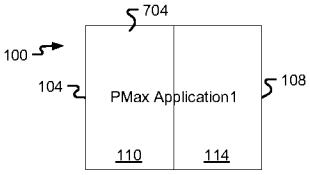


FIG. 9A

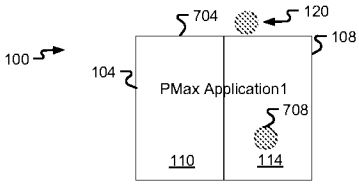


FIG. 9B

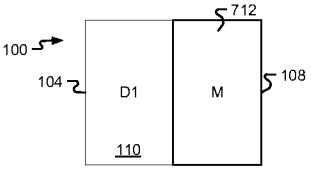


FIG. 9C

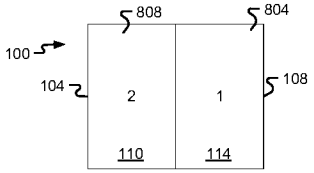


FIG. 10A

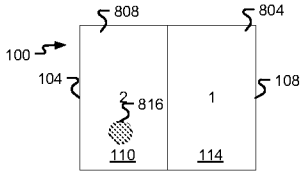


FIG. 10B

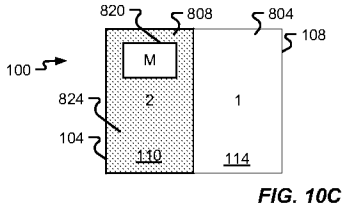
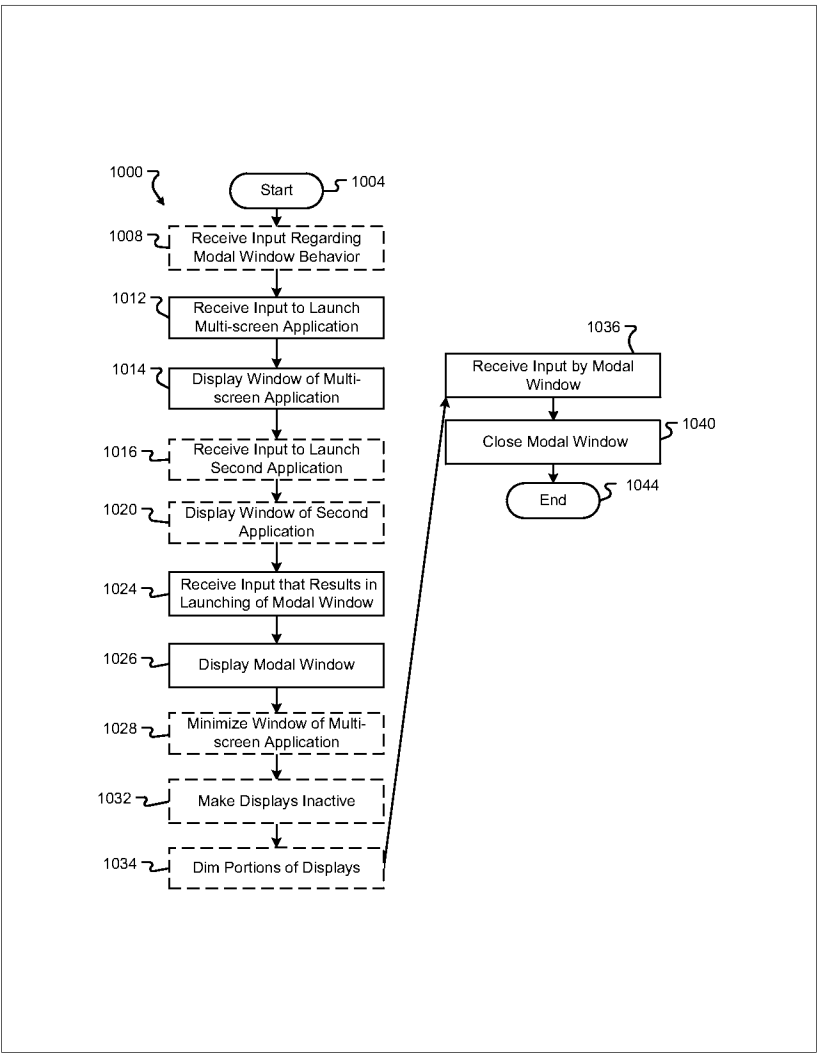


FIG. 10C


1000
Start
1004
1008
Receive Input Regarding Modal Window Behavior
1012
Receive Input to Launch Multi-screen Application
1014
Display Window of Multi-screen Application
1016
Receive Input to Launch Second Application
1020
Display Window of Second Application
1024
Receive Input that Results in Launching of Modal Window
1026
Display Modal Window
1028
Minimize Window of Multi-screen Application
1032
Make Displays Inactive
1034
Dim Portions of Displays
1036
Receive Input by Modal Window
1040
Close Modal Window
1044
End

FILLING STACK OPENING IN DISPLAY

[0100] The device 100 manages desktops and/or windows with at least one window stack 1700, 1728, as shown in Figs. 9A and 9B. A window stack 1700, 1728 is a logical arrangement of active and/or inactive windows for a multi-screen device. For example, the window stack 1700, 1728 may be logically similar to a deck of cards, where one or more windows or desktops are arranged in order, as shown in Figs. 9A and 9B. An active window is a window that is currently being displayed on at least one of the touch sensitive displays 110, 114. For example, windows 104 and 108 are active windows and are displayed on touch sensitive displays 110 and 114. An inactive window is a window that was opened and displayed but is now "behind" an active window and not being displayed. In embodiments, an inactive window may be for an application that is suspended, and thus, the window is not displaying active content. For example, windows 1712, 1716, 1720, and 1724 are inactive windows.

[0101] A window stack 1700, 1728 may have various arrangements or organizational structures. In the embodiment shown in Fig. 9A, the device 100 includes a first stack 1760 associated with a first touch sensitive display 110 and a second stack associated with a second touch sensitive display 114. Thus, each touch sensitive display 110, 114 can have an associated window stack 1760, 1764. These two window stacks 1760, 1764 may have different numbers of windows arranged in the respective stacks 1760, 1764. Further, the two window stacks 1760, 1764 can also be identified differently and managed separately. Thus, the first window stack 1760 can be arranged in order from a first window 1704 to a next window 1720 to a last window 1724 and finally to a desktop 1722, which, in embodiments, is at the "bottom" of the window stack 1760. In embodiments, the desktop 1722 is not always at the "bottom" as application windows can be arranged in the window stack below the desktop 1722, and the desktop 1722 can be brought to the "top" of a stack over other windows during a desktop reveal. Likewise, the second stack 1764 can be arranged from a first window 1708 to a next window 1712 to a last window 1716, and finally to a desktop 1718, which, in embodiments, is a single desktop area, with desktop 1722, under all the windows in both window stack 1760 and window stack 1764. A logical data structure for managing the two window stacks 1760, 1764 may be as described in conjunction with Fig. 10.

[0102] Another arrangement for a window stack 1728 is shown in Fig. 9B. In this embodiment, there is a single window stack 1728 for both touch sensitive displays 110, 114.

Thus, the window stack 1728 is arranged from a desktop 1758 to a first window 1744 to a last window 1756. A window can be arranged in a position among all windows without an association to a specific touch sensitive display 110, 114. In this embodiment, a window is in the order of windows. Further, at least one window is identified as being active. For example, a single window may be rendered in two portions 1732 and 1736 that are displayed on the first touch sensitive screen 110 and the second touch sensitive screen 114. The single window may only occupy a single position in the window stack 1728 although it is displayed on both displays 110, 114.

[0103] Yet another arrangement of a window stack 1760 is shown in Figs. 9C through 9E. The window stack 1760 is shown in three "elevation" views. In Fig. 9C, the top of the window stack 1760 is shown. Two sides of the window stack 1760 are shown in Figs. 9D and 9E. In this embodiment, the window stack 1760 resembles a stack of bricks. The windows are stacked on each other. Looking from the top of the window stack 1760 in Fig, 9C, only the top most windows in the window stack 1760 are seen in different portions of the composite display 1764. The composite display 1764 represents a logical model for the entire display area of the device 100, which can include touch sensitive display 110 and touch sensitive display 114. A desktop 1786 or a window can occupy part or all of the composite display 1764.

[0104] In the embodiment shown, the desktop 1786 is the lowest display or "brick" in the window stack 1760. Thereupon, window 1 1782, window 2 1782, window 3 1768, and window 4 1770 are layered. Window 1 1782, window 3 1768, window 2 1782, and window 4 1770 only occupy a portion of the composite display 1764. Thus, another part of the stack 1760 includes window 8 1774 and windows 5 through 7 shown in section 1790. Only the top window in any portion of the composite display 1764 is actually rendered and displayed. Thus, as shown in the top view in Fig. 9C, window 4 1770, window 8 1774, and window 3 1768 are displayed as being at the top of the display in different portions of the window stack 1760. A window can be dimensioned to occupy only a portion of the composite display 1760 to "reveal" windows lower in the window stack 1760. For example, window 3 1768 is lower in the stack than both window 4 1770 and window 8 1774 but is still displayed. A logical data structure to manage the window stack can be as described in conjunction with Fig. 10.

[0105] When a new window is opened, the newly activated window is generally positioned at the top of the stack. However, where and how the window is positioned within the stack can be

a function of the orientation of the device 100, the context of what programs, functions, software, etc. are being executed on the device 100, how the stack is positioned when the new window is opened, etc. To insert the window in the stack, the position in the stack for the window is determined and the touch sensitive display 110, 114 to which the window is associated may also be determined. With this information, a logical data structure for the window can be created and stored. When user interface or other events or tasks change the arrangement of windows, the window stack(s) can be changed to reflect the change in arrangement. It should be noted that these same concepts described above can be used to manage the one or more desktops for the device 100.

[0106] A logical data structure 1800 for managing the arrangement of windows or desktops in a window stack is shown in Fig. 10. The logical data structure 1800 can be any data structure used to store data whether an object, record, file, etc. The logical data structure 1800 can be stored in any type of database or data storage system, regardless of protocol or standard. In embodiments, the logical data structure 1800 includes one or more portions, fields, attributes, etc. that store data in a logical arrangement that allows for easy storage and retrieval of the information. Hereinafrer, these one or more portions, fields, attributes, etc. shall be described simply as fields. The fields can store data for a window identifier 1804, dimensions 1808, a stack position identifier 1812, a display identifier 1816, and/or an active indicator 1820. Each window in a window stack can have an associated logical data structure 1800. While only a single logical data structure 1800 is shown in Fig. 10, there may be more or fewer logical data structures 1800 used with a window stack (based on the number of windows or desktops in the stack), as represented by ellipses 1824. Further, there may be more or fewer fields than those shown in Fig. 10, as represented by ellipses 1828.

[0107] A window identifier 1804 can include any identifier (ID) that uniquely identifies the associated window in relation to other windows in the window stack. The window identifier 1804 can be a globally unique identifier (GUID), a numeric ID, an alphanumeric ID, or other type of identifier. In embodiments, the window identifier 1804 can be one, two, or any number of digits based on the number of windows that can be opened. In alternative embodiments, the size of the window identifier 1804 may change based on the number of windows opened. While the window is open, the window identifier 1804 may be static and remain unchanged.

[0108] Dimensions 1808 can include dimensions for a window in the composite display 1760. For example, the dimensions 1808 can include coordinates for two or more corners of the window or may include one coordinate and dimensions for the width and height of the window. These dimensions 1808 can delineate what portion of the composite display 1760 the window may occupy, which may the entire composite display 1760 or only part of composite display 1760. For example, window 4 1770 may have dimensions 1880 that indicate that the window 1770 will occupy only part of the display area for composite display 1760, as shown in Figs. 9C through 9E. As windows are moved or inserted in the window stack, the dimensions 1808 may change.

[0109] A stack position identifier 1812 can be any identifier that can identify the position in the stack for the window or may be inferred from the window's control record within a data structure, such as a list or a stack. The stack position identifier 1812 can be a GUID, a numeric ID, an alphanumeric ID, or other type of identifier. Each window or desktop can include a stack position identifier 1812. For example, as shown in Fig. 9A, window 1 1704 in stack 1 1760 can have a stack position identifier 1812 of 1 identifying that window 1704 is the first window in the stack 1760 and the active window. Similarly, window 6 1724 can have a stack position identifier 1812 of 3 representing that window 1724 is the third window in the stack 1760. Window 2 1708 can also have a stack position identifier 1812 of 1 representing that window 1708 is the first window in the second stack 1764. As shown in Fig. 9B, window 1 1744 can have a stack position identifier 1812 of 1, window 3, rendered in portions 1732 and 1736, can have a stack position identifier 1812 of 3, and window 6 1756 can have a stack position identifier 1812 of 6. Thus, depending on the type of stack, the stack position identifier 1812 can represent a window's location in the stack.

[0110] A display identifier 1816 can identify that the window or desktop is associated with a particular display, such as the first display 110 or the second display 114, or the composite display 1760 composed of both displays. While this display identifier 1816 may not be needed for a multi-stack system, as shown in Fig. 9A, the display identifier 1816 can indicate whether a window in the serial stack of Fig. 9B is displayed on a particular display. Thus, window 3 may have two portions 1732 and 1736 in Fig. 9B. The first portion 1732 may have a display identifier 1816 for the first display while the second portion 1736 may have a display identifier 1816 for the second display 114. However, in alternative embodiments, the window may have

two display identifier 1816 that represent that the window is displayed on both of the displays 110, 114, or a display identifier 1816 identifying the composite display. In another alternate embodiment, the window may have a single display identifier 1816 to represent that the window is displayed on both of the displays 110, 114.

[0111] Similar to the display identifier 1816, an active indicator 1820 may not be needed with the dual stack system of Fig. 9A, as the window in stack position 1 is active and displayed. In the system of Fig. 9B, the active indicator 1820 can indicate which window(s) in the stack is being displayed. Thus, window 3 may have two portions 1732 and 1736 in Fig. 9. The first portion 1732 may have an active indicator 1820 while the second portion 1736 may also have an active indicator 1820. However, in alternative embodiments, window 3 may have a single active indicator 1820. The active indicator 1820 can be a simple flag or bit that represents that the window is active or displayed.

[0112] An embodiment of a method 900 for creating a window stack is shown in Fig. 11. While a general order for the steps of the method 1900 is shown in Fig. 11. Generally, the method 1900 starts with a start operation 1904 and ends with an end operation 1928. The method 1900 can include more or fewer steps or can arrange the order of the steps differently than those shown in Fig. 11. The method 1900 can be executed as a set of computer-executable instructions executed by a computer system and encoded or stored on a computer readable medium. Hereinafrer, the method 1900 shall be explained with reference to the systems, components, modules, software, data structures, user interfaces, etc. described in conjunction with Figs. 1-10.

[0113] A multi-screen device 100 can receive activation of a window, in step 1908. In embodiments, the multi-screen device 100 can receive activation of a window by receiving an input from the touch sensitive display 110 or 114, the configurable area 112 or 116, a gesture capture region 120 or 124, or some other hardware sensor operable to receive user interface inputs. The processor may execute the Task Management Module 540 may receive the input. The Task Management Module 540 can interpret the input as requesting an application task to be executed that will open a window in the window stack.

[0114] In embodiments, the Task Management Module 540 places the user interface interaction in the task stack 552 to be acted upon by the Display Configuration Module 568 of the Multi-Display Management Module 524. Further, the Task Management Module 540 waits

for information from the Multi-Display Management Module 524 to send instructions to the Window Management Module 532 to create the window in the window stack.

[0115] The Multi-Display Management Module 524, upon receiving instruction from the Task Management Module 540, determines to which touch portion of the composite display 1760, the newly activated window should be associated, in step 1912. For example, window 4 1770 is associated with the a portion of the composite display 1764 In embodiments, the device state module 574 of the Multi-Display Management Module 524 may determine how the device is oriented or in what state the device is in, e.g., open, closed, portrait, etc. Further, the preferences module 572 and/or requirements module 580 may determine how the window is to be displayed. The gesture module 576 may determine the user's intentions about how the window is to be opened based on the type of gesture and the location of where the gesture is made.

[0116] The Display Configuration Module 568 may use the input from these modules and evaluate the current window stack 1760 to determine the best place and the best dimensions, based on a visibility algorithm, to open the window. Thus, the Display Configuration Module 568 determines the best place to put the window at the top of the window stack 1760, in step 1916. The visibility algorithm, in embodiments, determines for all portions of the composite display, which windows are at the top of the stack. For example, the visibility algorithm determines that window 3 1768, window 4 1770, and window 8 1774 are at the top of the stack 1760 as viewed in Figs. 9C through 9E. Upon determining where to open the window, the Display Configuration Module 568 can assign a display identifier 816 and possibly dimensions 808 to the window. The display identifier 816 and dimensions 808 can then be sent back to the Task Management Module 540. The Task Management Module 540 may then assign the window a stack position identifier 812 indicating the windows position at the top of the window stack.

[0117] In embodiments, the Task Management Module 540 sends the window stack information and instructions to render the window to the Window Management Module 532. The Window Management Module 532 and the Task Management Module 540 can create the logical data structure 800, in step 1924. Both the Task Management Module 540 and the Window Management Module 532 may create and manage copies of the window stack. These copies of the window stack can be synchronized or kept similar through communications between the Window Management Module 532 and the Task Management Module 540. Thus,

the Window Management Module 532 and the Task Management Module 540, based on the information determined by the Multi-Display Management Module 524, can assign dimensions 808, a stack position identifier 812 (e.g., window 1 1782, window 4 1770, etc.), a display identifier 816 (e.g., touch sensitive display 1 110, touch sensitive display 2 114, composite display identifier, etc,), and an active indicator 820, which is generally always set when the window is at the "top" of the stack. The logical data structure 800 may then be stored by both the Window Management Module 532 and the Task Management Module 540. Further, the Window Management Module 532 and the Task Management Module 540may thereinafrer manage the window stack and the logical data structure(s) 800.

[0118] FIG. 12 depicts a further window stacking configuration. A plurality of windows 1, 2, 3, 4, 5, 6, 7, and 8, whether from the same or different multi-screen or single-screen applications, are depicted. The touch sensitive display 110 currently has window 4 in the active display position while touch sensitive display 114 currently has window 5 in the active display position. The stack for touch sensitive display 110, from top to bottom, has window 4 in the active display position and windows 3, 2, and 1 positioned behind it. The stack for touch sensitive display 114, from top to bottom, has window 5 in the active display position and windows 6, 7, and 8 positioned behind it.

[0119] The desktops D1, D2, D3, D4, D5, and D6 are positioned behind the window stacks. The desktops can be viewed as a desktop stack different from the window stack. Viewed in this manner, the touch sensitive display 110 has a corresponding desktop stack comprising desktops D3, D2, and D1,with desktop D1 being at the bottom 2300 stack position and with desktop D3 at the top stack position being capable of being displayed with window 4 (depending on the window position and size (whether maximized or minimized), and the touch sensitive display 114 has a corresponding desktop stack has a corresponding desktop stack comprising desktops D4, D5, and D6,with desktop D6 being at the bottom 2304 stack position and with desktop D4 at the top stack position being capable of being displayed with window 5 (depending on the window position and size (whether maximized or minimized). Conceptually, the desktops can be viewed as a canvas divided, in this example, into six sections of which two can be displayed, at any one time, the touch sensitive displays 110, 114. When the device 100 is in the closed state, this conceptual model, in one configuration, persists. In this configuration, only one window and desktop stacks can be seen (that corresponding to the primary screen) but the other

window and desktop stacks are virtual; that is, they are maintained in memory but cannot be seen because the secondary screen is not enabled.

[0120] A number of examples will now be discussed with reference to closing windows and/or quitting executing or active applications. In the examples, it is understood that an application can itself be closed by a user, thereby causing the opened windows outputted by the application to close automatically. In other situations, a window of an application is closed by the user without causing the application itself to quit or terminate. In other situations, a window of an executing application is closed by a user, which automatically causes the application to quit or terminate. The examples discussed below are applicable to all of these variations.

[0121] A first example will be discussed with reference to Figs. 13A-C. The first touch sensitive display 110 displays a second window 2, and the second touch sensitive display 114 displays a third window 3. The windows are minimized in that they occupy only one of the primary and secondary screens 104 and 108. A first window 1 is in an inactive display position for a first window stack on the first touch sensitive display 110, while a fourth window 4 is in an inactive display position for a second window stack for the second touch sensitive display 114. Stated another way, the second window 2 is in an active display position for the first touch sensitive display 110, and the third window 3 is in an active display position for the second touch sensitive display 114. The first, second, third and fourth windows 1-4 correspond to different first, second, third, and fourth active (or executing) applications, which may be single- or multi-screen applications.

[0122] With reference to Fig. 13B, the user selects the second window 2 and quits or terminates the corresponding second application, leaving a gap behind in the first touch sensitive display 110 formerly occupied by the second window 2. The small rectangle 2 denotes the closing of the second window 2. As will be appreciated, when a window is closed, it ultimately disappears from view and is removed from the stack (as the corresponding application is terminated).

[0123] In one configuration shown in Fig. 18, a window 1800 is closed by a user. As shown by window 1804, the closed window gradually shrinks in size until it disappears from view; that is, the window 1800 of the application that was closed will fade out and scale in from approximately X =1 and Y = 1 (with X and Y each equal to one being the scale size of each of the first and second touch sensitive displays 110, 114) and approximately alpha = 1 (or 100%)

(the intensity, opacity, or brightness of the displayed image) (left view) gradually and progressively to a window 1804 of a smaller window size and area and lower alpha (middle view) (between alpha = 1 and alpha = 0) and finally gradually and progressively to a window 1808 having an even smaller scale size of approximately X = 0.5 and Y = 0.5 (or one-quarter of the area of the first touch sensitive display 110) and an alpha of approximately 0 (or 0%). At or after the window has decreased to the 0.5 X 0.5 scale size, the window disappears from view. The time duration from closing to disappearing of the closed window or from opening to full appearance of the opened window is typically from about 200 to about 1 sec and even more typically from about 250 to about 500 ms.

[0124] When an application is terminated by the user and the corresponding window(s) closed, the gap in a stack formerly occupied by the closed window(s) is filled by the window corresponding to a next active or executing application on the same display. As shown in Fig. 13C, the next window in the stack, namely the first window 1 corresponding to the first application, is now displayed by the first touch sensitive display 110. As shown in Fig. 18, the new window 1812, in this case the first window 1, has an initial scale size of X and Y each equal to approximately 2 (which is larger than the first touch sensitive display 110 being about four times its area) and alpha = 0 and gradually and progressively decreases to a smaller size, with the final scale size of window 1816 being approximately X = 1 and Y = 1, or the size of the first touch sensitive display 110, and alpha = 1 (or the final displayed image intensity, opacity, or brightness). A first desktop D1 has moved into the top inactive display position for the first touch sensitive display 110. The second touch sensitive display 114 is not changed during the operation as the third application corresponding to the third window 3 is active and executing.

[0125] In one configuration, the moving and closing of the windows in Fig. 18 occur substantially simultaneously. With continuing reference to Fig. 18, at a start time the user closes the window1800. The window then gradually and progressively diminishes in size and alpha value until the end time at which the closed window disappears from view. Also, at the start time, the window 1812 is moved into the touch sensitive display area and has a larger size than the display area. Its alpha value is approximately zero, meaning that it is not visible. The window 1812 gradually and progressively increases in size and alpha value until the end time at which the new window 1816 is in full view.

[0126] In other embodiments, the moving and closing of the windows in Fig. 18 do not occur substantially simultaneously. For example, the closing of the window precedes the moving of the new window into the display area vacated by the closed window.

[0127] In other embodiments, the increasing and decreasing in the size and/or alpha value of a closing or moving window is not done gradually or progressively but can be done abruptly, discretely, discontinuously, stepwise, or in any other manner.

[0128] A second example will be applied to Figs. 14A-C. The first touch sensitive and second touch sensitive displays 110, 114 have the same configurations as discussed above with reference to Fig. 13A-C.

[0129] With reference to Fig. 14B, the user selects the third window 3 and quits or terminates the corresponding third application, leaving a gap behind in the second touch sensitive display 114 formerly occupied by the third window 2. The small rectangle 3 denotes the closing of the third window 3. As shown in Fig. 14C, the next window in the stack, namely the fourth window 4 corresponding to the fourth application, is now displayed by the second touch sensitive display 114. A second desktop D2 has moved into the top inactive display position for the second touch sensitive display 114. The first touch sensitive display 110 is not changed during the operation as the second application corresponding to the second window 2 is active and executing.

[0130] In the above examples, when there is no window corresponding to an active application remaining in the stack corresponding to the touch sensitive display of a closed window or terminated application, the underlying desktop is moved into the active display position. In the case of Fig. 13C, the first desktop D1 would be displayed if the first window 1 and first application were to be closed. In the case of Fig. 14C, the second desktop D2 would be displayed if the fourth window 4 and fourth application were to be closed.

[0131] With reference to Figs. 15A-C, a further example will be discussed. While the device 100 of Figs. 13A-C and 14A-C is in dual screen mode in which both touch sensitive displays are enabled and displaying content, the device 100 of Figs. 15A-C are in single screen mode in which only one touch sensitive display is enabled and displaying content. Stated another way, the device 100 is closed in Figs. 15A-C and opened in Figs. 13A-C and 14A-C. The enabled first touch sensitive display 110 corresponding to the first screen 104 is displaying, in Fig. 15A, a first window corresponding to a first active application. The stack further includes a second window 2 corresponding to a second active application, a third window 3 corresponding to a

third active application, and a first desktop D1. The user closes the first window 1 and thereby terminates the first application.

[0132] With reference to Fig. 15B, the first window disappears from view, leaving a gap in the stack.

[0133] With reference to Fig. 15C, the second window 2 corresponding to the second application moves to the active display position and is now displayed by the first touch sensitive display 110. The third window 3 corresponding to the third application is now in at the top inactive display position in stack.

[0134] The gap is not filled from the stack (not shown) corresponding to the inactive second touch sensitive display 114. Referring to Fig. 12, for instance, the first touch sensitive display 110 has first, second, third, and fourth windows 1, 2, 3, and 4 corresponding to first, second, and third applications in its stack, and the second touch sensitive display 114 has fifth, sixth, seventh, and eighth windows 5, 6, 7, and 8 corresponding to fifth, sixth, seventh, and eighth applications in its stack. When the device 100 is closed, leaving only the first touch sensitive display 110 active and the second touch sensitive display 114 inactive, stack openings are filled from the first touch sensitive display 110 stack and not the second touch sensitive display 114 stack. Thus, if the fourth window were to be closed, the first touch sensitive display 110 begins displaying the third window 3 and not the fifth, sixth, seventh, or eighth windows 5, 6, 7, and 8. In other configurations, the opposite is true; that is, the if the fourth window were to be closed, the first touch sensitive display 110 would begins displaying the fifth window 5 and not the first, second, or third windows 1, 2, or 3.

[0135] A further example will now be discussed with reference to Figs. 16A-C. Before discussing Figs. 16A-C, windows in multi-screen applications will be discussed with reference to Fig. 8. As shown in Fig. 8, first window 1.1 is a root window, second window 1.1.1 is a node window, and third window 1.1.1.1 is a leaf window. The windows are in a hierarchical tree-like sequence and are controlled by a common multi-screen application. The root window 1.1 may be a top level view of the hierarchical application window sequence such that there is no parent window corresponding to the root window. The root window may be a parent to a node window. One or more node windows may be provided that are related as parent/children. A node window may also serve as a leaf window. By leaf window, it is meant that the leaf window has no corresponding node screen for which the leaf window is a parent. As such, the leaf window does

not have any children node windows, Windows 1.1.1.1 and 1.1.2 are examples of leaf windows. The root view refers to all of the windows controlled by a multi-screen application at a point in time.

[0136] Referring to Fig. 16A, the device 100 is in dual screen mode, or is open, and is displaying (maximized) first and second root view windows 1.1.1 and 1.1.2 corresponding to a (maximized) first active multi-screen application. The first and second related windows 1.1.1 and 1.1.2 are root, node, or leaf windows related, such as by a sibling relationship in which the windows 1.1.1 and 1.1.2 have a common parent or root window. The stack for the first touch sensitive display 110 has a second window 2 for a second active single-screen application in its first stack, while the second touch sensitive display 114 has a third window 3 for a third active single-screen application in its second stack. The user closes the first application, causing the first and second related windows 1.1.1 and 1.1.2 to disappear from view and the first and second stacks. The top inactive display position occupied by the second window 2 is moved to the active display position for the first touch sensitive display 110, and the top inactive display position occupied by the third window is moved to the active display position for the second touch sensitive display 114.

[0137] A multi-screen application remembers the last-seen windows within its internal data structures. For instance, the multi-screen application, when relaunched, will relaunch with the first and second related windows 1.1.1 and 1.1.2 being presented to the user. In other configurations, the multi-screen application, when relaunched, relaunches in a single-screen mode, even when closed in a maximized state, such as that shown in Fig. 16A. In that event, the window presented to the user upon relaunch is the parent view of the two windows seen upon quitting. In the example of Fig. 16A, the parent window of windows 1.1.1 and 1.1.2 is a window 1.1. Expose views reemerge with their parent windows.

[0138] In one embodiment, when an application is closed and there is no window in the stack, the desktop appears. The desktop remains still in the background and is not animated. This can happen when a window of a single-screen application is closed, when a maximized dual window multi-screen application is closed (and there is no open window in either stack of the dual screens), or when a maximized multi-screen application is closed (and there is one open window in one of the dual screens but not in the other). In the latter case, the desktop will appear in one of the dual screens and the open window in the other.

[0139] In the various examples, middleware 520, particularly one or more of the Multi-Display Management (MDM) class 524, a Surface Cache class 528, a Window Management class 532, an Activity Management class 536, and an Application Management class 540, independently or collectively, receive a command to close a window and a corresponding application, such as by detecting a gesture (step 1700). In response, middleware 520 determines whether the stack in the affected touch sensitive display has an active application and window in the stack (step 1704). When a window is in the stack (step 1708), the middleware 520 fades the closed window out and fades the window in the stack in to the touch sensitive display (step 1712). When a window is not in the stack (step 1708), the middleware 520 fades a closed window out into the desktop (step 1716).

[0140] The exemplary systems and methods of this disclosure have been described in relation to communication devices. However, to avoid unnecessarily obscuring the present disclosure, the preceding description omits a number of known structures and devices. This omission is not to be construed as a limitation of the scopes of the claims. Specific details are set forth to provide an understanding of the present disclosure. It should however be appreciated that the present disclosure may be practiced in a variety of ways beyond the specific detail set forth herein.

[0141] Furthermore, while the exemplary aspects, embodiments, and/or configurations illustrated herein show the various components of the system collocated, certain components of the system can be located remotely, at distant portions of a distributed network, such as a LAN and/or the Internet, or within a dedicated system. Thus, it should be appreciated, that the components of the system can be combined in to one or more devices, such as a communication device, or collocated on a particular node of a distributed network, such as an analog and/or digital telecommunications network, a packet-switch network, or a circuit-switched network. It will be appreciated from the preceding description, and for reasons of computational efficiency, that the components of the system can be arranged at any location within a distributed network of components without affecting the operation of the system. For example, the various components can be located in a switch such as a PBX and media server, gateway, in one or more communications devices, at one or more users' premises, or some combination thereof. Similarly, one or more functional portions of the system could be distributed between a telecommunications device(s) and an associated computing device.

[0142] Furthermore, it should be appreciated that the various links connecting the elements can be wired or wireless links, or any combination thereof, or any other known or later developed element(s) that is capable of supplying and/or communicating data to and from the connected elements. These wired or wireless links can also be secure links and may be capable of communicating encrypted information. Transmission media used as links, for example, can be any suitable carrier for electrical signals, including coaxial cables, copper wire and fiber optics, and may take the form of acoustic or light waves, such as those generated during radio-wave and infra-red data communications.

[0143] Also, while the flowcharts have been discussed and illustrated in relation to a particular sequence of events, it should be appreciated that changes, additions, and omissions to this sequence can occur without materially affecting the operation of the disclosed embodiments, configuration, and aspects.

[0144] A number of variations and modifications of the disclosure can be used. It would be possible to provide for some features of the disclosure without providing others.

[0145] In other embodiments, other rules may be applied. For example, the closed window may be replaced by an open window of an active application located at the bottom of or some other intermediate position in the stack.

[0146] In yet another embodiment, the systems and methods of this disclosure can be implemented in conjunction with a special purpose computer, a programmed microprocessor or microcontroller and peripheral integrated circuit element(s), an ASIC or other integrated circuit, a digital signal processor, a hard-wired electronic or logic circuit such as discrete element circuit, a programmable logic device or gate array such as PLD, PLA, FPGA, PAL, special purpose computer, any comparable means, or the like. In general, any device(s) or means capable of implementing the methodology illustrated herein can be used to implement the various aspects of this disclosure. Exemplary hardware that can be used for the disclosed embodiments, configurations and aspects includes computers, handheld devices, telephones (e.g., cellular, Internet enabled, digital, analog, hybrids, and others), and other hardware known in the art. Some of these devices include processors (e.g., a single or multiple microprocessors), memory, nonvolatile storage, input devices, and output devices. Furthermore, alternative software implementations including, but not limited to, distributed processing or component/object

distributed processing, parallel processing, or virtual machine processing can also be constructed to implement the methods described herein.

[0147] In yet another embodiment, the disclosed methods may be readily implemented in conjunction with software using object or object-oriented software development environments that provide portable source code that can be used on a variety of computer or workstation platforms. Alternatively, the disclosed system may be implemented partially or fully in hardware using standard logic circuits or VLSI design. Whether software or hardware is used to implement the systems in accordance with this disclosure is dependent on the speed and/or efficiency requirements of the system, the particular function, and the particular software or hardware systems or microprocessor or microcomputer systems being utilized.

[0148] In yet another embodiment, the disclosed methods may be partially implemented in software that can be stored on a storage medium, executed on programmed general-purpose computer with the cooperation of a controller and memory, a special purpose computer, a microprocessor, or the like. In these instances, the systems and methods of this disclosure can be implemented as program embedded on personal computer such as an applet, JAVA® or CGI script, as a resource residing on a server or computer workstation, as a routine embedded in a dedicated measurement system, system component, or the like. The system can also be implemented by physically incorporating the system and/or method into a software and/or hardware system.

[0149] Although the present disclosure describes components and functions implemented in the aspects, embodiments, and/or configurations with reference to particular standards and protocols, the aspects, embodiments, and/or configurations are not limited to such standards and protocols. Other similar standards and protocols not mentioned herein are in existence and are considered to be included in the present disclosure. Moreover, the standards and protocols mentioned herein and other similar standards and protocols not mentioned herein are periodically superseded by faster or more effective equivalents having essentially the same functions. Such replacement standards and protocols having the same functions are considered equivalents included in the present disclosure.

[0150] The present disclosure, in various aspects, embodiments, and/or configurations, includes components, methods, processes, systems and/or apparatus substantially as depicted and described herein, including various aspects, embodiments, configurations embodiments,

subcombinations, and/or subsets thereof. Those of skill in the art will understand how to make and use the disclosed aspects, embodiments, and/or configurations afrer understanding the present disclosure. The present disclosure, in various aspects, embodiments, and/or configurations, includes providing devices and processes in the absence of items not depicted and/or described herein or in various aspects, embodiments, and/or configurations hereof, including in the absence of such items as may have been used in previous devices or processes, e.g., for improving performance, achieving ease and\or reducing cost of implementation.

[0151] The foregoing discussion has been presented for purposes of illustration and description. The foregoing is not intended to limit the disclosure to the form or forms disclosed herein. In the foregoing Detailed Description for example, various features of the disclosure are grouped together in one or more aspects, embodiments, and/or configurations for the purpose of streamlining the disclosure. The features of the aspects, embodiments, and/or configurations of the disclosure may be combined in alternate aspects, embodiments, and/or configurations other than those discussed above. This method of disclosure is not to be interpreted as reflecting an intention that the claims require more features than are expressly recited in each claim. Rather, as the following claims reflect, inventive aspects lie in less than all features of a single foregoing disclosed aspect, embodiment, and/or configuration. Thus, the following claims are hereby incorporated into this Detailed Description, with each claim standing on its own as a separate preferred embodiment of the disclosure.

[0152] Moreover, though the description has included description of one or more aspects, embodiments, and/or configurations and certain variations and modifications, other variations, combinations, and modifications are within the scope of the disclosure, e.g., as may be within the skill and knowledge of those in the art, afrer understanding the present disclosure. It is intended to obtain rights which include alternative aspects, embodiments, and/or configurations to the extent permitted, including alternate, interchangeable and/or equivalent structures, functions, ranges or steps to those claimed, whether or not such alternate, interchangeable and/or equivalent structures, functions, ranges or steps are disclosed herein, and without intending to publicly dedicate any patentable subject matter.

What is claimed is:

1. A method, comprising:

receiving, by processor executable middleware, a command to close a first window associated with an active first application, the window being displayed by a touch sensitive display;

when a second window from a different active second application is in a stack of the touch sensitive display, ceasing, by the processor executable middleware, the display of the first window and commencing, by the processor executable middleware, the display of the second window; and

when no second window from a different active second application is in the stack, ceasing, by the processor executable middleware, the display of the first window and commencing, by the processor executable middleware, the display of a desktop.

2. The method of claim 1, wherein the touch sensitive display comprises first and second touch sensitive displays, wherein the first window is displayed on the first touch sensitive display but not by the second touch sensitive display, wherein the second touch sensitive display displays a third window, and further comprising:

maintaining, by the processor executable middleware, display of the third window during closing of the first window.

3. The method of claim 1, wherein the second window is in the stack of the touch sensitive display, wherein the second window is at a top-most or bottom-most inactive display position in the stack, and wherein the processor executable middleware commences the display of the second window on the touch sensitive display.

4. The method of claim 1, wherein no second window is in the stack and wherein the processor executable middleware commences the display of the desktop on the touch sensitive display.

5. The method of claim 4, wherein the touch sensitive display comprises first and second touch sensitive displays having first and second stacks, respectively, wherein the first window is displayed on the first touch sensitive display but not by the second touch sensitive display, wherein the second touch sensitive display displays a third window, and wherein the first stack has no second window and the second stack has a fourth window.

6. The method of claim 1, wherein the touch sensitive display comprises first and second touch sensitive displays having first and second stacks, respectively, wherein the first and second touch sensitive displays display at least one of first and second portions of the first window and first and third windows, respectively, wherein the first application is a multi-screen application, and wherein the following rules are applied:

when a second and fourth window from at least a different active second application are in the first and second stacks, respectively, ceasing, by the processor executable middleware, the display of the at least one of first and second portions of the first window and first and third windows and commencing, by the processor executable middleware, the display of the second and fourth windows by the first and second touch sensitive displays, respectively;

when no second and fourth window from a different active second application is in the first and second stacks, respectively, ceasing, by the processor executable middleware, the display of the at least one of first and second portions of the first window and first and third windows and commencing, by the processor executable middleware, the display of a desktop in each of the first and second touch sensitive displays; and

when no second window from a different active second application is in the first stack but a fourth window from a different active second application is in the second stack, ceasing, by the processor executable middleware, the display of the at least one of first and second portions of the first window and first and third windows and commencing, by the processor executable middleware, the display of a desktop in the first touch sensitive display and the desktop in the second touch sensitive display.

7. The method of claim 6, wherein, in the ceasing steps, the first window gradually and/or progressively decreases in size and at least one of intensity, opacity, and brightness until the first window disappears from view.

8. A non-transient, tangible computer readable storage medium comprising processor executable instructions operable to perform the steps of claim 1.

9. A communication device, comprising:

at least first and second touch sensitive displays to display windows;

a processor; and

a computer readable memory, the computer readable memory comprising processor executable middleware operable to:

receive a command to close a first window associated with an active first application and displayed by the first touch sensitive display;

when a second window from a different active second application is in a stack of the first touch sensitive display, cease the display of the first window by the first touch sensitive display and commence the display of the second window by the first touch sensitive display; and

when no second window from a different active second application is in the stack, cease the display of the first window by the first touch sensitive display and commence the display of a desktop by the first touch sensitive display.

10. The device of claim 9, wherein the second touch sensitive display displays a third window and wherein the processor executable middleware is operable to maintain display of the third window during closing of the first window.

11. The device of claim 9, wherein the second window is at a top-most or bottom-most inactive display position in the stack and wherein the processor executable middleware commences the display of the second window on the first touch sensitive display.

12. The device of claim 9, wherein no second window is in the stack and wherein the processor executable middleware commences the display of the desktop on the first touch sensitive display.

13. The device of claim 12, wherein the first and second touch sensitive displays have first and second stacks, respectively, wherein the first window is displayed on the first touch sensitive display but not by the second touch sensitive display, wherein the second touch sensitive display displays a third window, and wherein the first stack has no second window and the second stack has a fourth window.

14. The device of claim 9, wherein the first and second touch sensitive displays have first and second stacks, respectively, wherein the first and second touch sensitive displays display at least one of first and second portions of the first window and first and third windows, respectively, wherein the first application is a multi-screen application, and wherein the following rules are applied:

when a second and fourth window from at least a different active second application are in the first and second stacks, respectively, ceasing, by the processor executable middleware, the display of the at least one of first and second portions of the first window and first and third

windows and commencing, by the processor executable middleware, the display of the second and fourth windows by the first and second touch sensitive displays, respectively;

when no second and fourth window from a different active second application is in the first and second stacks, respectively, ceasing, by the processor executable middleware, the display of the at least one of first and second portions of the first window and first and third windows and commencing, by the processor executable middleware, the display of a desktop in each of the first and second touch sensitive displays; and

when no second window from a different active second application is in the first stack but a fourth window from a different active second application is in the second stack, ceasing, by the processor executable middleware, the display of the at least one of first and second portions of the first window and first and third windows and commencing, by the processor executable middleware, the display of a desktop in the first touch sensitive display and the desktop in the second touch sensitive display.

15. The method of claim 14, wherein, in the ceasing operation, the first window gradually and/or progressively decreases in size and at least one of intensity, opacity, and brightness until the first window disappears from view.

16. A method, comprising:

receiving a command to quit an application, thereby closing a window outputted by the application; and

gradually and/or progressively decreasing a scale size and/or alpha value of the window, whereby the window disappears from view.

17. The method of claim 16, wherein the scale size of the window is gradually and/or progressively decreased.

18. The method of claim 16, wherein the alpha value of the window is gradually and/or progressively decreased.

19. The method of claim 16, wherein the closed window is to be replaced by a second open window, and further comprising:

gradually and/or progressively decreasing a scale size of the second window and/or increasing an alpha value of the second window, whereby the window appears in view.

20. The method of claim 19, wherein the scale size of the second window is gradually and/or progressively decreased and wherein the gradually and/or progressively decreasing steps occur substantially simultaneously.

21. The method of claim 19, wherein an alpha value of the second window is gradually and/or progressively increased and wherein the gradually and/or progressively decreasing steps occur substantially simultaneously.

22. A communication device comprising dual touch sensitive displays operable to perform the steps of claim 16.

23. A non-transient and tangible computer readable storage medium comprising processor executable instructions to perform the steps of claim 16.

ABSTRACT

[0154] The present disclosure is directed to methodologies and devices for quitting an active application and controlling displayed images as a result of quitting the application.

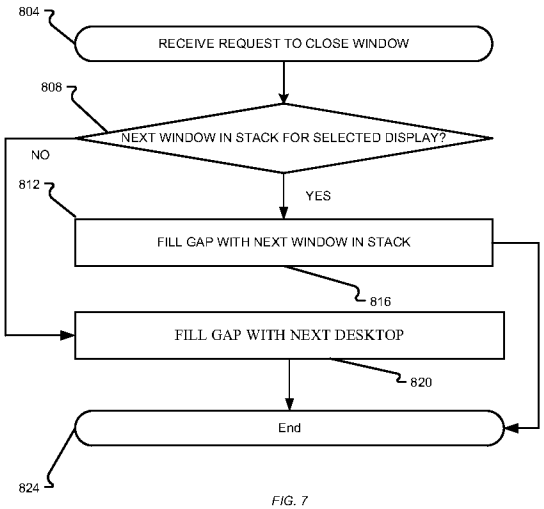


FIG. 7

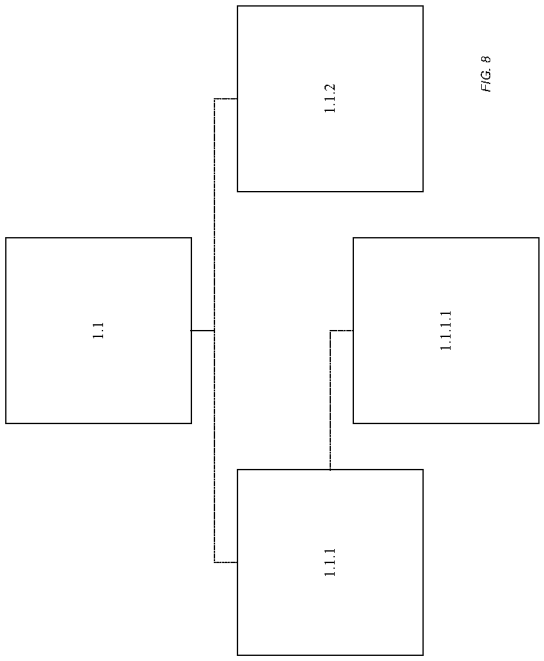


FIG. 8

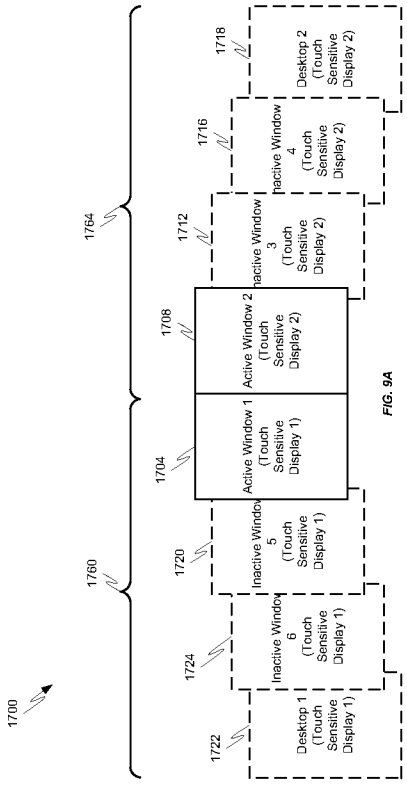


FIG. 9A

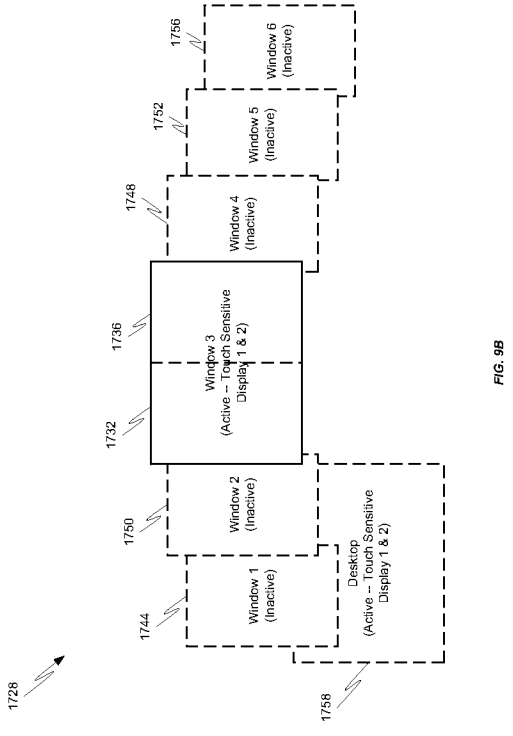


FIG. 9B

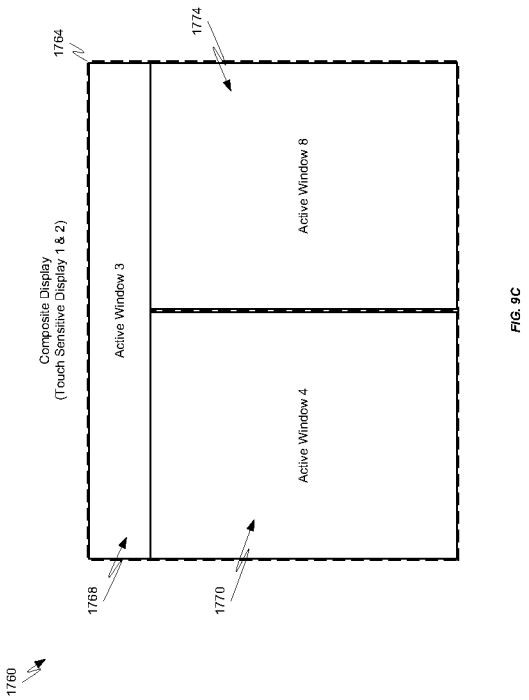


FIG. 9C

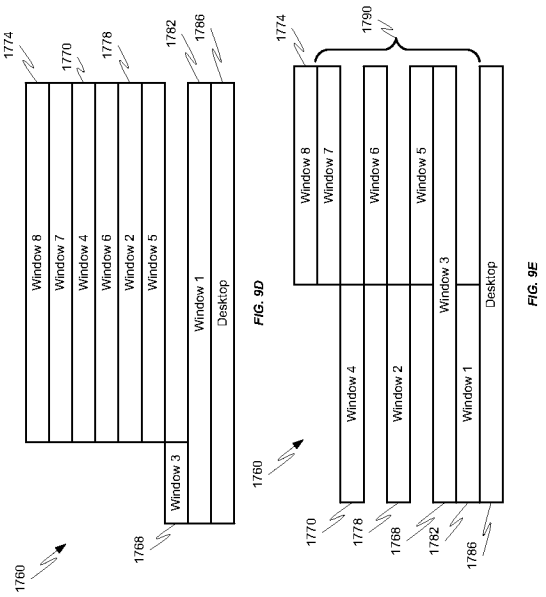


FIG. 9D

FIG. 9E

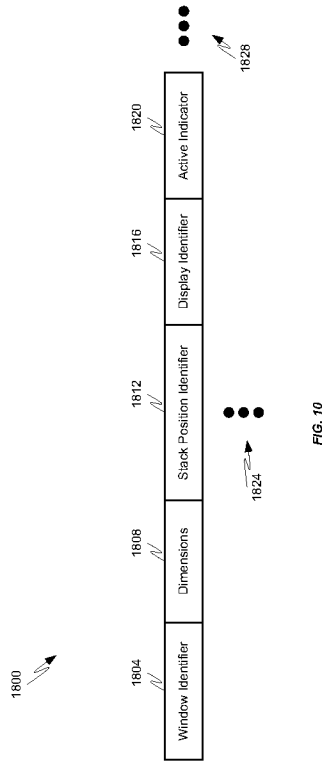


FIG. 10

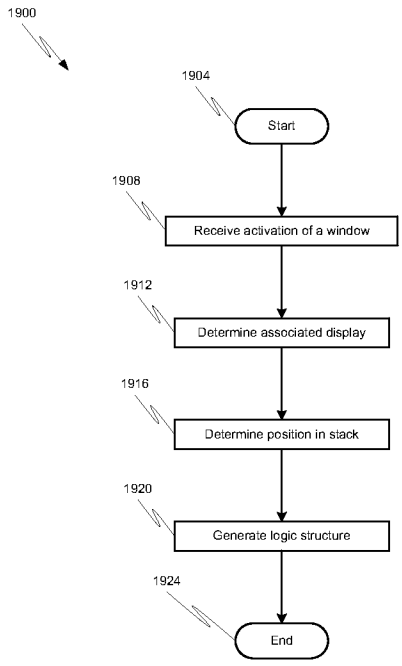


FIG. 11

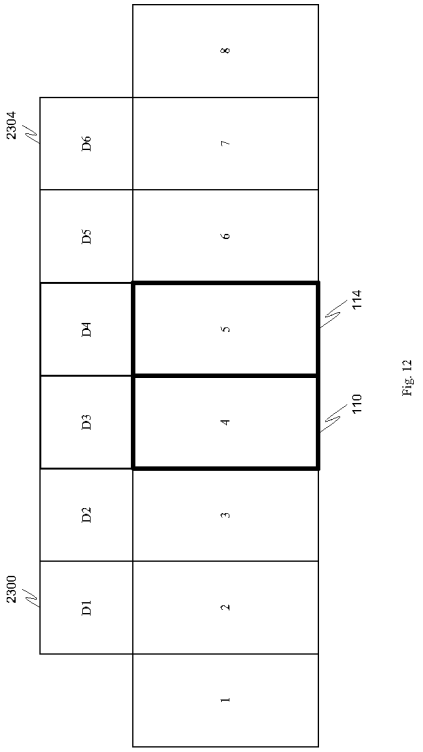


Fig. 12

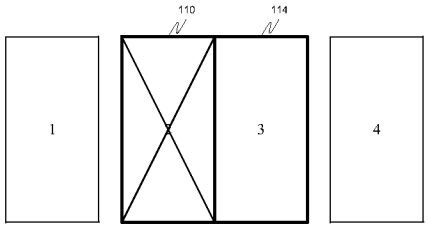


FIG. 13A

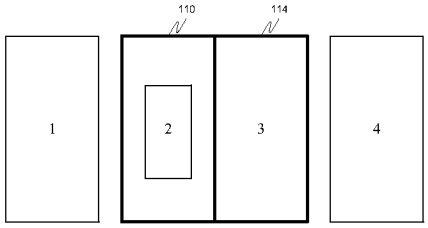


FIG. 13B

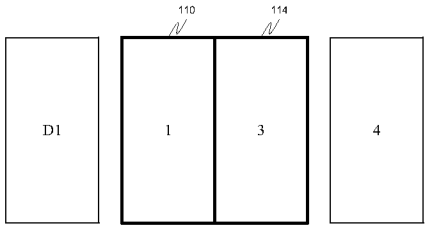


FIG. 13C

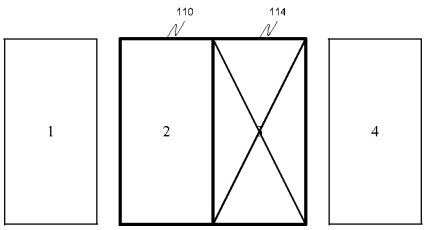


FIG. 14A

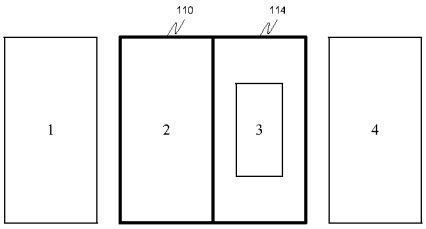


FIG. 14B

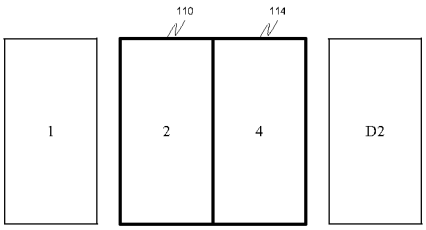


FIG. 14C

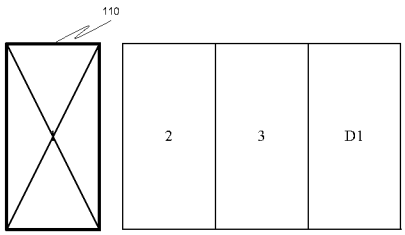


FIG. 15A

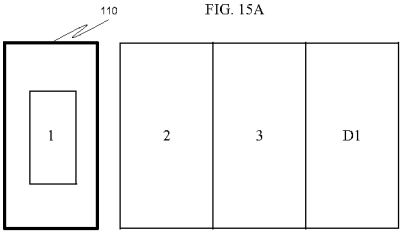


FIG. 15B

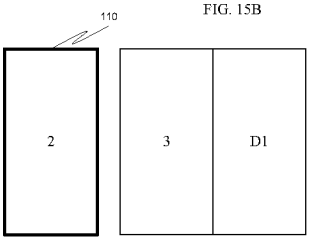


FIG. 15C

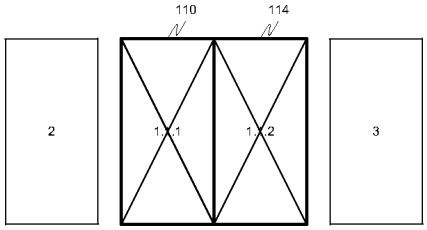


FIG. 16A

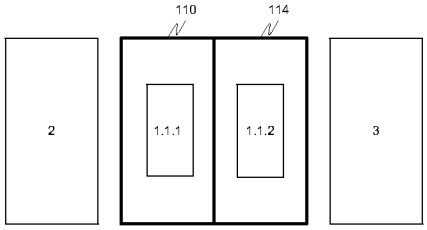


FIG. 16B

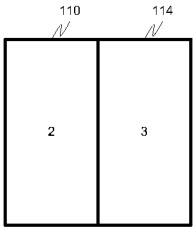


FIG. 16C

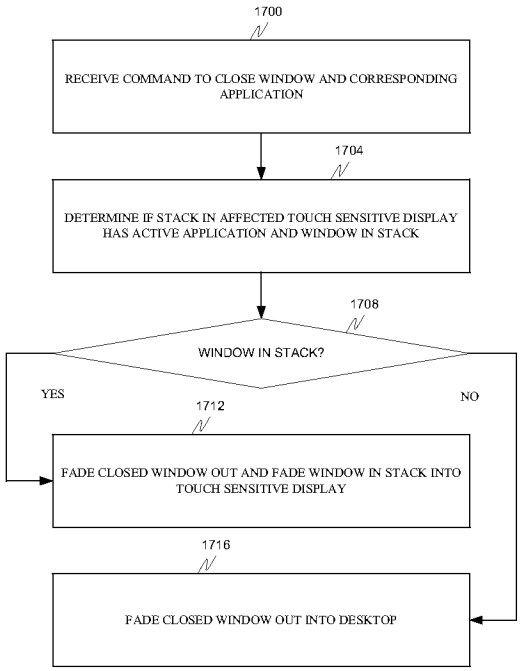


FIG. 17

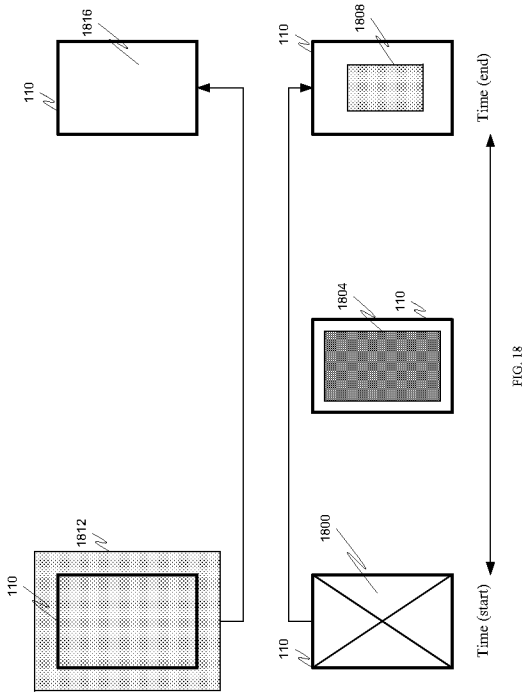


FIG. 18

[0001] Brief Description of Drawings

[0002] FIG. 7 is a first representation of an embodiment of a device configuration generated in response to an independent display orientation module;

[0003] FIG. 8 is a second representation of an embodiment of a device configuration generated in response to the independent display orientation module ;

[0004] FIG. 9 is a third representation of an embodiment of a device configuration generated in response to the independent display orientation module; and

[0005] FIG. 10 is a fourth representation of an embodiment of a device configuration generated in response to the independent display orientation module.

Detailed Description

[0006] Firmware and Software:

[0007] The memory 508 may store and the processor 504 may execute one or more software components. These components can include at least one operating system (OS) 516, an application manager 562, a desktop 566, and/or one or more applications 564a and/or 564b from an application store 560. The OS 516 can include a framework 520, one or more frame buffers 548, one or more drivers 512, previously described in conjunction with Fig. 2, and/or a kernel 518. The OS 516 can be any software, consisting of programs and data, which manages computer hardware resources and provides common services for the execution of various applications 564. The OS 516 can be any operating system and, at least in some embodiments, dedicated to mobile devices, including, but not limited to, Linux, ANDROID TM, iPhone OS (IOS TM), WINDOWS PHONE 7 TM, etc. The OS 516 is operable to provide functionality to the phone by executing one or more operations, as described herein.

[0008] The applications 564 can be any higher level software that executes particular functionality for the user. Applications 564 can include programs such as email clients, web browsers, texting applications, games, media players, office suites, etc. The applications 564 can be stored in an application store 560, which may represent any memory or data storage, and the management software associated therewith, for storing the applications 564. Once executed, the applications 564 may be run in a different area of memory 508.

[0009] The framework 520 may be any sofrware or data that allows the multiple tasks running on the device to interact. In embodiments, at least portions of the framework 520 and the discrete components described hereinafrer may be considered part of the OS 516 or an application 564. However, these portions will be described as part of the framework 520, but those components are not so limited. The framework 520 can include, but is not limited to, a Multi-Display Management (MDM) module 524, a Surface Cache module 528, a Window Management module 532, an Input Management module 536, a Task Management module 540, an Independent Display Orientation (IDO) module 587, an Application Model Manager 542, a Display Controller, one or more frame buffers 548, a task stack 552, one or more window stacks 550 (which is a logical arrangement of windows and/or desktops in a display area), and/or an event buffer 556.

[0010] The MDM module 524 includes one or more modules that are operable to manage the display of applications or other data on the screens of the device. An embodiment of the MDM module 524 is described in conjunction with Fig. 5B. In embodiments, the MDM module 524 receives inputs from the other OS 516 components, such as, the drivers 512, and from the applications 564 to determine continually the state of the device 100. The inputs assist the MDM module 524 in determining how to configure and allocate the displays according to the application's preferences and requirements, and the user's actions. Once a determination for display configurations is made, the MDM module 524 can bind the applications 564 to a display. The configuration may then be provided to one or more other components to generate a window with a display.

[0011] The Surface Cache module 528 includes any memory or storage and the software associated therewith to store or cache one or more images of windows. A series of active and/or non-active windows (or other display objects, such as, a desktop display) can be associated with each display. An active window (or other display object) is currently displayed. A non-active windows (or other display objects) were opened and, at some time, displayed but are now not displayed. To enhance the user experience, before a window transitions from an active state to an inactive state, a "screen shot" of a last generated image of the window (or other display object) can be stored. The Surface Cache module 528 may be operable to store a bitmap of the last active image of a window (or other display object) not currently displayed. Thus, the Surface Cache module 528 stores the images of non-active windows (or other display objects) in a data store.

[0012] In embodiments, the Window Management module 532 is operable to manage the windows (or other display objects) that are active or not active on each of the displays. The Window Management module 532, based on information from the MDM module 524, the OS 516, or other components, determines when a window (or other display object) is visible or not active. The Window Management module 532 may then put a non-visible window (or other display object) in a "not active state" and, in conjunction with the Task Management module Task Management 540 suspends the application's operation. Further, the Window Management module 532 may assign, through collaborative interaction with the MDM module 524, a display identifier to the window (or other display object) or manage one or more other items of data associated with the window (or other display object). The Window Management module 532 may also provide the stored information to the application 564, the Task Management module 540, or other components interacting with or associated with the window (or other display object). The Window Management module 532 can also associate an input task with a window based on window focus and display coordinates within the motion space.

[0013] The Input Management module 536 is operable to manage events that occur with the device. An event is any input into the window environment, for example, a user interface interactions with a user. The Input Management module 536 receives the events and logically stores the events in an event buffer 556. Events can include such user interface interactions as a "down event," which occurs when a screen 104, 108 receives a touch signal from a user, a "move event," which occurs when the screen 104, 108 determines that a user's finger is moving across a screen(s), an "up event, which occurs when the screen 104, 108 determines that the user has stopped touching the screen 104, 108, etc. These events are received, stored, and forwarded to other modules by the Input Management module 536. The Input Management module 536 may also map screen inputs to a motion space which is the culmination of all physical and virtual display available on the device.

[0014] The IDO module 587 includes one or more modules that are operable to further manage the display of applications or other data on the screens of the device in conjunction with the MDM module 524 and Requirement Module 580. Specifically, the IDO module 587 simultaneously manages inputs from the applications 564 which may require specific orientations and inputs generated from user which may require a different orientation as between two different applications displayed on the respective screens 104 and 108. For example, in the case of running the "Angry Birds" application that requires the landscape orientation, a user may wish to run an application for display on the other screen in a portrait orientation. Accordingly, the IDO module 587 handles inputs from different applications and/or inputs generated by the user in order to display the applications on a selective screen-by-screen basis.

[0015] Figures 7 and 8 depict two different output configurations of the device 100 with the same applications/desktops running on the respective screens. Specifically, Fig. 7 depicts the device 100 being in the Dual-Portrait (PD) output configuration where data is displayed on both the primary screen 104 and the secondary screen 108. In this example, the device 100 displays an application A1 on a window 1102 in a first portrait configuration 1106, and an application A2 on window 1104 in a second portrait configuration 1108.

[0016] Referring to Figure 8, the output configuration has changed for the window 1102 running the application A1. As shown, the application A1 has now been displayed in a landscape configuration 1107. In this case, the IDO module 587 has responded to either a user input or an input from the application A1, or both, in order to display the application A1 in the landscape configuration while the application A2 is still displayed in the portrait configuration 1108.

[0017] Figures 9 and 10 depict additional output configurations of the device 100 with the same applications/desktops running on the respective screens. Specifically, Fig. 9 depicts the device 100 being in the Dual-Landscape (LD) output configuration where data is displayed on both the primary screen 104 and the secondary screen 108. In this example, the device 100 displays the application A1 again on the window or window 1102 but in a first landscape configuration 1110, and the application A2 is again on the window 1104 but in a second landscape configuration 1112.

[0018] Referring to Figure 10, the output configuration has changed for the window 1102 running the application A1. As shown, the application A1 has now been displayed in a portrait configuration 1111. The IDO module 587 again has responded to either a user input or an input from the application A1, or both, in order to display the application A1 in the portrait configuration, while the application A2 is still displayed in the landscape configuration 1112.

[0019] There are a number of advantages of providing the capability to independently control the orientation of the desktops or applications displayed on separate screens. In the case of an application like "Angry Birds" that requires a landscape configuration, the user still has the ability to selectively display the desktop or application running on the other screen. Further, the user is able to selectively lock one or both of the displays on the screens in desired orientations despite the physical orientation of the device. Further for example, the user may wish to keep a particular application in a landscape or portrait configuration despite the physical orientation of the device that may otherwise automatically change the orientation of the displays based on inputs from the position sensors 172 and/or the accelerometer 176. The user may execute one or more input gestures either on off-screen areas or on-screen areas of the device in order to selectively change, lock, or unlock the display orientations. A user may further set one or more predetermined controls for determining what takes precedence in terms of displaying and then modifying the orientation of the displays. For example, a user could set a first control protocol in which the orientation of all desktops/applications would only be displayed in accordance with specific user inputs, or the user could set another control protocol in which all desktops/applications would first be displayed based on other inputs such as application inputs or device inputs, and then displayed based on subsequent user inputs.

Claims

[0020] 1. A method of controlling data displayed by a multi-screen device, comprising:

determining that the multi-screen device has been placed in a dual display state with a first desktop or application displayed on a first display and second desktop or application displayed on a second display, wherein data from the first and second desktops/application are displayed on their respective first and second displays;

determining that the first and second desktops or applications are each displayed in respective first portrait or landscape orientations;

determining that an input is received from at least one of the desktops or applications, or from the device, or from a user, instructing that the first orientations of one of the desktops or applications should change to a second different orientation; and

in response to determining that an input is received, causing the data from a selected one of the first or second desktops or applications to be displayed in the second different orientation.

[0021] 2. The method of claim 1, wherein the first and second desktops or applications are displayed in first portrait orientations, and in response to determining that the input is received, causing the data from a selected one of the first or second desktops or applications to be displayed in a landscape orientation as the second different orientation, and the non-selected one of the first or second desktops remains in the first portrait orientation.

[0022] 3. The method of claim 1, wherein the first and second desktops or applications are displayed in first landscape orientations, and in response to determining that the input is received, causing the data from a selected one of the first or second desktops or applications to be displayed in a portrait orientation as the second different orientation, and the non-selected one of the first or second desktops remains in the first landscape orientation.

[0023] 4. The method of claim 1, wherein the input is at least one of a user input, application input, or device input comprising data received from a position sensor or an accelerometer.

[0024] 5. The method of claim 1, wherein a user sets a control protocol to determine precedence in which input from a plurality of inputs controls the data from the selected one of the first or second desktops or applications to be displayed in the second different orientation.

[0025] 6. The method of claim 4, wherein the at least one input comprises a plurality of inputs including application, device, and user inputs.

[0026] 7. The method of Claim 1, further including:

determining that an input is received from the user instructing that the first orientation of one of the desktops or applications should be locked, and in response to determining that the user input is received, causing the data from another selected one of the first or second desktops or applications to remain displayed in the first orientation while the data from the selected one of the first or second desktops is displayed in the second different orientation.

[0027] 8. A non-transitory computer-readable medium having stored thereon instructions that cause a computing system to execute a method, the instructions comprising:

instructions configured to determine that the multi-screen device has been placed in a dual display state with a first desktop or application displayed on a first display and second desktop or application displayed on a second display, wherein data from the first and second desktops/application are displayed on their respective first and second displays;

instructions configured to determine that the first and second desktops or applications are each displayed in respective first portrait or landscape orientations;

instructions configured to determine that an input is received from at least one of the desktops or applications, or from the device, or from a user, instructing that the first orientations of one of the desktops or applications should change to a second different orientation; and

in response to determining that an input is received, instructions configured to cause the data from a selected one of the first or second desktops or applications to be displayed in the second different orientation.

[0028] 9. The non-transitory computer readable medium of claim 8, wherein the first and second desktops or applications are displayed in first portrait orientations, and in response to determining that the input is received, causing the data from a selected one of the first or second desktops or applications to be displayed in a landscape orientation as the second different orientation, and the non-selected one of the first or second desktops remains in the first portrait orientation.

[0029] 10. The non-transitory computer readable medium of claim 8, wherein the first and second desktops or applications are displayed in first landscape orientations, and in response to determining that the input is received, causing the data from a selected one of the first or second desktops or applications to be displayed in a portrait orientation as the second different orientation, and the non-selected one of the first or second desktops remains in the first landscape orientation.

[0030] 11. The non-transitory computer readable medium of claim 8, wherein the input is at least one of a user input, application input, or device input comprising data received from a position sensor or an accelerometer.

[0031] 12. The non-transitory computer readable medium of claim 8, wherein a user sets a control protocol to determine precedence in which input from a plurality of inputs controls the data from the selected one of the first or second desktops or applications to be displayed in the second different orientation.

[0032] 13. The non-transitory computer readable medium of claim 8, wherein the at least one input comprises a plurality of inputs including application, device, and user inputs.

[0033] 14. The non-transitory computer readable medium of claim 8, further including:

determining that an input is received from the user instructing that the first orientation of one of the desktops or applications should be locked, and in response to determining that the user input is received, causing the data from another selected one of the first or second desktops or applications to remain displayed in the first orientation while the data from the selected one of the first or second desktops is displayed in the second different orientation.

[0034] 15. A multi-screen user device, comprising:

a first display including a first display area;

a second display including a second display area;

an independent display orientation element configured to manage a plurality of inputs corresponding to data to determine the orientation of desktops or applications displayed on the first and second displays; and

a computer-readable medium having instructions stored thereon that include:

a first set of instructions configured to determine that the multi-screen device has been placed in a dual display state with a first desktop or application displayed on a first display and second desktop or application displayed on a second display, wherein data from the first and second desktops/application are displayed on their respective first and second displays;

a second set of instructions configured to determine that the first and second desktops or applications are each displayed in respective first portrait or landscape orientations;

a third set of instructions configured to determine that an input is received from at least one of the desktops or applications, or from the device, or from a user, instructing that the first orientations of one of the desktops or applications should change to a second different orientation; and

in response to determining that an input is received, a fourth set of instructions configured to cause the data from a selected one of the first or second desktops or applications to be displayed in the second different orientation.

[0035] 16. The multi-screen device of claim 15, wherein the first and second desktops or applications are displayed in first portrait orientations, and in response to determining that the input is received, causing the data from a selected one of the first or second desktops or applications to be displayed in a landscape orientation as the second different orientation, and the non-selected one of the first or second desktops remains in the first portrait orientation.

[0036] 17. The multi-screen device of claim 15, wherein the first and second desktops or applications are displayed in first landscape orientations, and in response to determining that the input is received, causing the data from a selected one of the first or second desktops or applications to be displayed in a portrait orientation as the second different orientation, and the non-selected one of the first or second desktops remains in the first landscape orientation.

[0037] 18. The multi-screen device of claim 15, wherein a user sets a control protocol to determine precedence in which input from a plurality of inputs controls the data from the selected one of the first or second desktops or applications to be displayed in the second different orientation.

[0038] 19. The multi-screen device of claim 15, wherein the at least one input comprises a plurality of inputs including application, device, and user inputs.

[0039] 20. The multi-screen device of claim 15 further including:

a fourth set of instructions to determine that an input is received from the user instructing that the first orientation of one of the desktops or applications should be locked, and in response to determining that the user input is received, causing the data from another selected one of the first or second desktops or applications to remain displayed in the first orientation while the data from the selected one of the first or second desktops is displayed in the second different orientation.

ABSTRACT

A dual-screen user device and methods for revealing a combination of selected desktops and applications on single and dual screens are disclosed. Desktops and applications can be shifted between screens by user gestures, and/or moved off of the screens and therefore hidden. Hidden desktops and screens can be re-displayed by other gestures. The desktops and applications are arranged in a window stack that represents a logical order of the desktops and applications providing a user with an intuitive ability to manage multiple applications/desktops running simultaneously. One embodiment provides the user with the ability to selectively change the orientation of one display and leaving the other display unaffected when the user has two applications or desktops displayed on respective screens. The user can therefore selectively change on a screen-by-screen basis the orientation of any desktops or applications displayed.

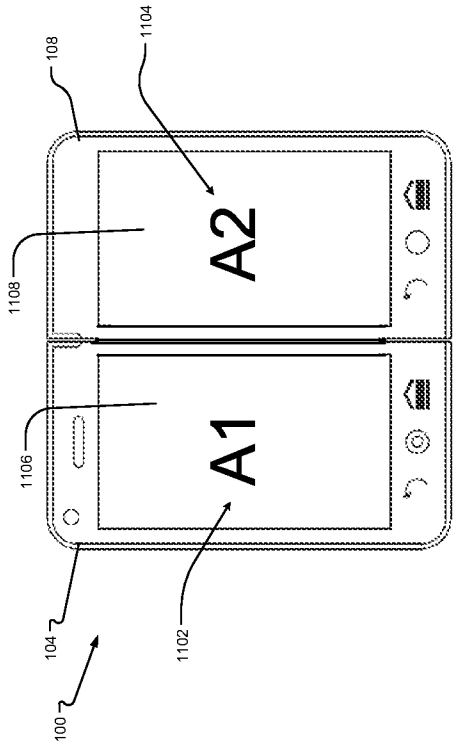


Fig. 7

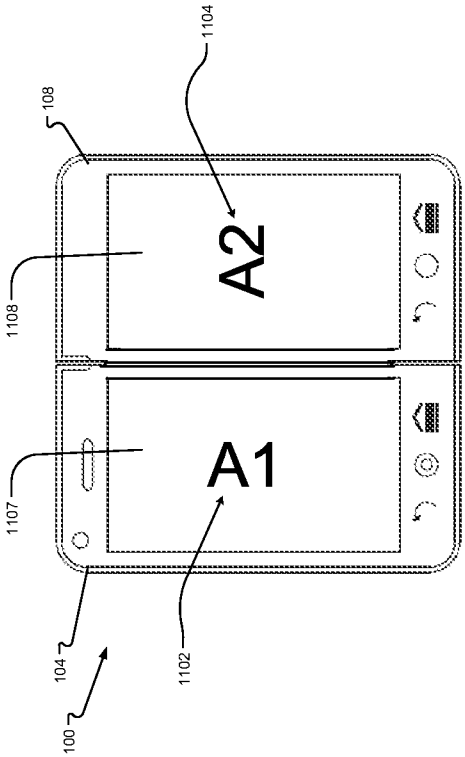


Fig. 8

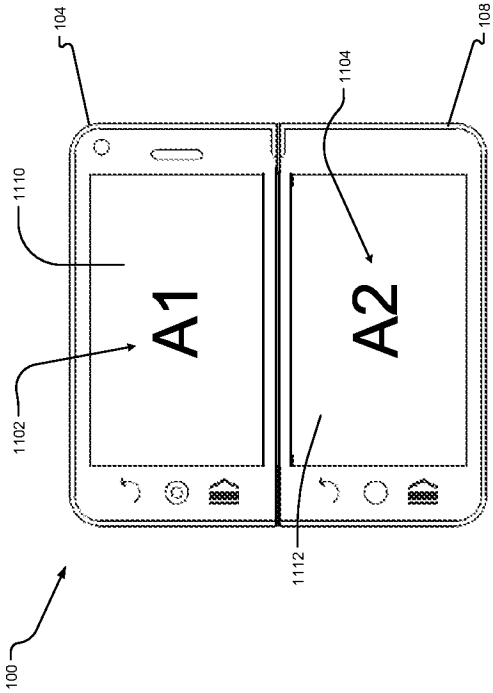


Fig. 9

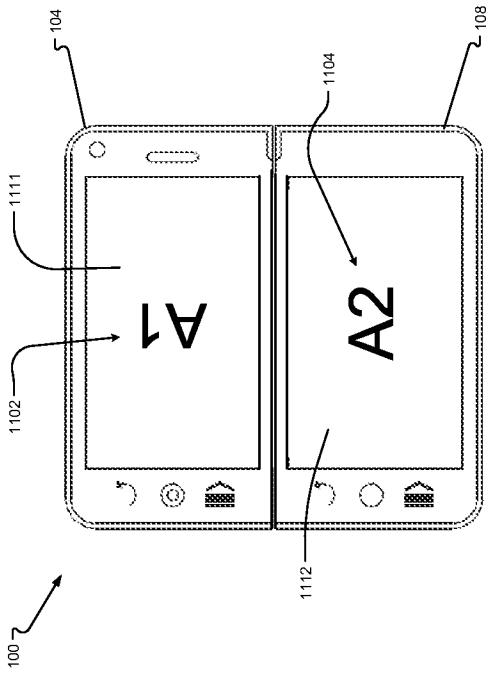


Fig. 10

ROTATION GRAVITY DROP

[0001] Display Controls:

[0002] Figs. 7A and 7B depict two examples of gesture sequences that can cause the device 100 to alter the display of data thereon. Specifically, Fig. 7A depicts a first gesture sequence 700 which results in a first display action. Although the first display action is depicted as being performed on the touch sensitive displays 110, 114, it should be appreciated that the first display action can alternatively, or additionally, be performed on other parts of the primary screen 104 and/or secondary screen 108. The first gesture sequence 700 begins when the device 100 is in an open portrait state 320. While in this state, the device 100 displays data on both its displays 110, 114. In some embodiments, the device 100 may display application data from different applications (e.g., a first and second application) on the displays 110, 114. In the depicted example, a first set of data from a first application may initially be displayed in a first portrait configuration 604 and a second set of data from the first application may be initially displayed in a second portrait configuration 608. In other words, the first and second displays 110, 114 may depict first and second aspects of a first application that has been maximized and is being displayed in both displays 110, 114 in either a PD output configuration or a PMax output configuration. It should be appreciated, however, that one of the displays may be depicting a desktop rather than data from an application. Also, it may be possible that one of the displays is depicting data from a first application whereas the other display is depicting data from a second application.

[0003] The first gesture sequence 700 may continue when a first user input 704 is received that causes the device 100 to rotate. In some embodiments, the device 100 may be rotated in the direction of arrow 708 (e.g., clockwise) by 90 degrees, although such a rotation is not required. Rather, a counter-clockwise rotation by 90 degrees or any rotation by a predetermined amount greater than or less than 90 degrees may also qualify as the first user input 704. The rotation of the device 100 may be detected, in some embodiments, by registering a rotation event with the task management module 540 which subsequently informs the MDM module 524 of the rotation. In some embodiments, the rotation event is registered by receiving one or more inputs from the accelerometer 176. During the rotation, the displays 110, 114 may continue to display the same data as before the first input 704 was received.

[0004] A second user input 712 may cause the rotation of the device 100 to stop. In some embodiments, the device 100 may be oriented in an open landscape state 348 after the second user input 712 is received. At this point, the MDM module 524 may determine the relative position of the displays 110, 114 (e.g., determine which display or screen 104, 108 is above the other) and based on the relative position of the displays in the final orientation, the MDM module 524 may instruct the window management module 532 to cause data from the application in the upper display (display 110 in this example) to "drop" down onto the lower display. In the example depicted in Fig. 7A, the first set of data from the first application may be automatically maximized across both displays 110, 114. This automated dropping of the data from the upper display may be referred to as a gravity drop of the data displayed because the data is "dropped" under the force of gravity as detected by the accelerometer 176.

[0005] As discussed above, the data originally displayed by the display that eventually becomes the upper display may be obtained from an application. This application may correspond to a single-screen or multi-screen application. If the application is a single-screen application, then the application may be automatically maximized into an Lmax output configuration. If the application is a multi-screen application, then the application can be automatically maximized into either an Lmax output configuration or an LD output configuration, depending upon the maximization rules for that application. Accordingly, the output of the first gesture sequence 700 may correspond to a display of application data simultaneously in first and second landscape configurations 612, 616 or as an Lmax output configuration 628.

[0006] Fig. 7B depicts a second gesture sequence 714 which results in a second display action. Like the first gesture sequence 700, the second gesture sequence 714 can be performed on any component of the device 100. The second gesture sequence 714 begins when device 100 is in an open landscape state 348. While in this state, the device 100 displays data on both its displays 110, 114. In some embodiments, the device 100 may be displaying data from a first application on both the displays 110, 114 in either an LD output configuration or an Lmax output configuration. The manner in which the application is displayed on both displays 110, 114 may depend upon the maximization rules for that application as determined by the MDM module 524.

[0007] The second gesture sequence 714 continues when the MDM module 524 registers a first user input 716 that corresponds to a rotation of the device 100. Again, the input registered

at the MDM module 524 may correspond to a data input provided by the task management module 540 in response to the task management module 540 detecting the physical rotation of the device 100. In some embodiments, the device 100 may be rotated in a direction of arrow 720 (e.g., counter-clockwise) by 90 degrees, although such a rotation is not required. Rather, a clockwise rotation by 90 degrees or any rotation by a predetermined amount greater or less than 90 degrees may also qualify as the first user input 716. During the rotation, the displays 110, 114 may continue to display the same data as before the first input 716 was received.

[0008] A second user input 724 may cause the rotation of the device 100 to stop. In some embodiments, the device 100 may be oriented in an open portrait state 320 after the second user input 724 is received. At this point, the MDM module 524 and/or window management module 532 may determine which application was originally being displayed in the open landscape state 348 and the window management module 532 may determine the maximization rules for that application. Specifically, if the application corresponds to a multi-screen application, the application may be kept in a maximized state, since it was originally in a maximized state. However, the application may be re-laid out based on its internal structure; in the case of a multi-screen application, standard maximization rules may be referenced which cause the application to be displayed in a PD output configuration. However, if the maximization rules dictate that Pmax output configuration 624 is required for the application belonging to the top screen, then the window management module 532 may cause the application to be displayed in a Pmax output configuration 624.

[0009] Referring now to Fig. 8, a first display control method 800 will be described in accordance with at least some embodiments of the present disclosure. The method 800 is initiated at step 804. Specifically, a device 100 may be initially in an open portrait state 320. The method continues when the rotation of the device 100 is detected (step 808). Specifically, the transition of the device 100 from an open portrait state 320 to an open landscape state 348 may be detected in step 808.

[0010] The method continues with the MDM module 524 and/or window management module 532 determining whether the application that was being presented in the now upper display was in a maximized state (step 812). Specifically, the MDM module 524 and/or window management module 532 may determine the state of the application (e.g., whether maximized or minimized) when the device 100 was in the open portrait state 320 and based on that

determination, decide how to automatically re-render the display of the application in the open landscape state 348.

[0011] If the application was maximized before the device 100 was rotated from an open portrait state 320 to an open landscape state 348, then the window management module 532 may execute a gravity drop on the application such that the application is automatically maximized across both displays 110, 114 in the landscape state 348 (step 816). In some embodiments, the application that is now on the upper screen/display may be automatically maximized across both displays 110, 114 in either an LD output configuration or an Lmax output configuration.

[0012] If the application was not maximized before the device 100 was rotated, then the window management module 532 may reference the minimization display rules for the application and may display the application on its original screen without performing a gravity drop (step 820). In particular, if a desktop or a different application was originally displayed on the now bottom display, then that same data or a variation thereof may be displayed on the bottom display. The application on the top display may still be displayed on the top display, but the data from that application may be reformatted for display in a landscape output configuration rather than a portrait output configuration.

[0013] Referring now to Fig. 9, a second display control method 900 will be described in accordance with at least some embodiments of the present disclosure. The method 900 is initiated at step 904. Specifically, a device 100 may be initially in an open landscape state 348. The method continues when the rotation of the device 100 is detected (step 908). Specifically, the transition of the device 100 from an open landscape state 348 to an open portrait state 320 may be detected in step 908.

[0014] The method continues with the MDM module 524 and/or window management module 532 determining whether the application that was being presented in the upper display was in a maximized state (step 912). Specifically, the MDM module 524 and/or window management module 532 may determine whether a gravity drop was previously imposed on the display of a multi-screen application and, if so, the window management module 532 may cause the application to be re-formatted for display in a portrait output configuration (step 916). Specific portrait-maximization rules may be referenced by the window management module 532 for the application in question. The window management module 532 may also determine that some of the data from the application should be displayed on one of the displays 110 in a first portrait

configuration 604, whereas other data from the same application should be displayed on the other display 114 in a second portrait configuration 608.

[0015] If the application was not maximized before the rotation of the device 100 to a portrait state, the window management module 532 may reference minimization rules to determine how the application should be displayed by the device 100 (step 920). In particular, the window management module 532 may determine whether the application should be displayed on one or two displays in the minimized state.

[0016] The exemplary systems and methods of this disclosure have been described in relation to display control mechanisms. However, to avoid unnecessarily obscuring the present disclosure, the preceding description omits a number of known structures and devices. This omission is not to be construed as a limitation of the scopes of the claims. Specific details are set forth to provide an understanding of the present disclosure. It should however be appreciated that the present disclosure may be practiced in a variety of ways beyond the specific detail set forth herein.

[0017] Furthermore, while the exemplary aspects, embodiments, and/or configurations illustrated herein show the various components of the system collocated, certain components of the system can be located remotely, at distant portions of a distributed network, such as a LAN and/or the Internet, or within a dedicated system. Thus, it should be appreciated, that the components of the system can be combined in to one or more devices, such as a computer, laptop, netbook, smart phone, mobile device, etc., or collocated on a particular node of a distributed network, such as an analog and/or digital telecommunications network, a packet-switch network, or a circuit-switched network. It will be appreciated from the preceding description, and for reasons of computational efficiency, that the components of the system can be arranged at any location within a distributed network of components without affecting the operation of the system. For example, the various components can be located in a switch such as a PBX and media server, gateway, in one or more communications devices, at one or more users' premises, or some combination thereof. Similarly, one or more functional portions of the system could be distributed between a telecommunications device(s) and an associated computing device.

[0018] Furthermore, it should be appreciated that the various links connecting the elements can be wired or wireless links, or any combination thereof, or any other known or later developed

element(s) that is capable of supplying and/or communicating data to and from the connected elements. These wired or wireless links can also be secure links and may be capable of communicating encrypted information. Transmission media used as links, for example, can be any suitable carrier for electrical signals, including coaxial cables, copper wire and fiber optics, and may take the form of acoustic or light waves, such as those generated during radio-wave and infra-red data communications.

[0019] Also, while the flowcharts have been discussed and illustrated in relation to a particular sequence of events, it should be appreciated that changes, additions, and omissions to this sequence can occur without materially affecting the operation of the disclosed embodiments, configuration, and aspects.

[0020] A number of variations and modifications of the disclosure can be used. It would be possible to provide for some features of the disclosure without providing others.

[0021] In yet another embodiment, the systems and methods of this disclosure can be implemented in conjunction with a special purpose computer, a programmed microprocessor or microcontroller and peripheral integrated circuit element(s), an ASIC or other integrated circuit, a digital signal processor, a hard-wired electronic or logic circuit such as discrete element circuit, a programmable logic device or gate array such as PLD, PLA, FPGA, PAL, special purpose computer, any comparable means, or the like. In general, any device(s) or means capable of implementing the methodology illustrated herein can be used to implement the various aspects of this disclosure. Exemplary hardware that can be used for the disclosed embodiments, configurations and aspects includes computers, handheld devices, telephones (e.g., cellular, Internet enabled, digital, analog, hybrids, and others), and other hardware known in the art. Some of these devices include processors (e.g., a single or multiple microprocessors), memory, nonvolatile storage, input devices, and output devices. Furthermore, alternative software implementations including, but not limited to, distributed processing or component/object distributed processing, parallel processing, or virtual machine processing can also be constructed to implement the methods described herein.

[0022] In yet another embodiment, the disclosed methods may be readily implemented in conjunction with software using object or object-oriented software development environments that provide portable source code that can be used on a variety of computer or workstation platforms. Alternatively, the disclosed system may be implemented partially or fully in

hardware using standard logic circuits or VLSI design. Whether software or hardware is used to implement the systems in accordance with this disclosure is dependent on the speed and/or efficiency requirements of the system, the particular function, and the particular software or hardware systems or microprocessor or microcomputer systems being utilized.

[0023] In yet another embodiment, the disclosed methods may be partially implemented in software that can be stored on a storage medium, executed on programmed general-purpose computer with the cooperation of a controller and memory, a special purpose computer, a microprocessor, or the like. In these instances, the systems and methods of this disclosure can be implemented as program embedded on personal computer such as an applet, JAVA® or CGI script, as a resource residing on a server or computer workstation, as a routine embedded in a dedicated measurement system, system component, or the like. The system can also be implemented by physically incorporating the system and/or method into a software and/or hardware system.

[0024] Although the present disclosure describes components and functions implemented in the aspects, embodiments, and/or configurations with reference to particular standards and protocols, the aspects, embodiments, and/or configurations are not limited to such standards and protocols. Other similar standards and protocols not mentioned herein are in existence and are considered to be included in the present disclosure. Moreover, the standards and protocols mentioned herein and other similar standards and protocols not mentioned herein are periodically superseded by faster or more effective equivalents having essentially the same functions. Such replacement standards and protocols having the same functions are considered equivalents included in the present disclosure.

[0025] The present disclosure, in various aspects, embodiments, and/or configurations, includes components, methods, processes, systems and/or apparatus substantially as depicted and described herein, including various aspects, embodiments, configurations embodiments, subcombinations, and/or subsets thereof. Those of skill in the art will understand how to make and use the disclosed aspects, embodiments, and/or configurations afrer understanding the present disclosure. The present disclosure, in various aspects, embodiments, and/or configurations, includes providing devices and processes in the absence of items not depicted and/or described herein or in various aspects, embodiments, and/or configurations hereof,

including in the absence of such items as may have been used in previous devices or processes, e.g., for improving performance, achieving ease and\or reducing cost of implementation.

[0026] The foregoing discussion has been presented for purposes of illustration and description. The foregoing is not intended to limit the disclosure to the form or forms disclosed herein. In the foregoing Detailed Description for example, various features of the disclosure are grouped together in one or more aspects, embodiments, and/or configurations for the purpose of streamlining the disclosure. The features of the aspects, embodiments, and/or configurations of the disclosure may be combined in alternate aspects, embodiments, and/or configurations other than those discussed above. This method of disclosure is not to be interpreted as reflecting an intention that the claims require more features than are expressly recited in each claim. Rather, as the following claims reflect, inventive aspects lie in less than all features of a single foregoing disclosed aspect, embodiment, and/or configuration. Thus, the following claims are hereby incorporated into this Detailed Description, with each claim standing on its own as a separate preferred embodiment of the disclosure.

[0027] Moreover, though the description has included description of one or more aspects, embodiments, and/or configurations and certain variations and modifications, other variations, combinations, and modifications are within the scope of the disclosure, e.g., as may be within the skill and knowledge of those in the art, afrer understanding the present disclosure. It is intended to obtain rights which include alternative aspects, embodiments, and/or configurations to the extent permitted, including alternate, interchangeable and/or equivalent structures, functions, ranges or steps to those claimed, whether or not such alternate, interchangeable and/or equivalent structures, functions, ranges or steps are disclosed herein, and without intending to publicly dedicate any patentable subject matter.

What is claimed is:

1. A method of controlling data displayed by a multi-screen device, comprising:

determining that the multi-screen device has been rotated from a first open state to a second different open state;

determining a relative position of a first screen that originally displayed data from a first application when the multi-screen device was in the first open state and a second screen that was not displaying data from the first application when the multi-screen device was in the first open state; and

controlling data displayed on the second screen in the second open state based on the determined relative position of the first screen and second screen.

2. The method of claim 1, wherein the first open state corresponds to an open portrait state and wherein the second open state corresponds to an open landscape state in which the first screen is positioned above the second screen.

3. The method of claim 2, wherein the rotation of the device is determined, at least in part, based on data received from an accelerometer.

4. The method of claim 2, further comprising:

determining whether the first application is a single-screen application or a multi-screen application; and

wherein controlling data displayed on the second screen in the open landscape state is also based on whether the first application is a single-screen application or a multi-screen application.

5. The method of claim 4, wherein the first application is a multi-screen application, the method further comprising:

automatically causing data from the first application to be displayed on the second screen in the open landscape state.

6. The method of claim 5, wherein data from the first application is displayed on both the first and second screens in the open landscape state in at least one of a dual-landscape output configuration and a landscape max output configuration.

7. The method of claim 4, wherein the first application is a single-screen application and wherein data from a second application is displayed on the second screen.

8. The method of claim 1, wherein the first state corresponds to an open landscape state and wherein the second state corresponds to an open portrait state.

9. The method of claim 8, wherein data from the first application is displayed on both the first and second screens in the first state, wherein data from the first application is automatically re-formatted to be displayed only on the first screen in the second state, and wherein data from a second application is automatically displayed on the second screen simultaneous with data from the first application being displayed on the first screen in the second state.

10. A non-transitory computer-readable medium having stored thereon instructions that cause a computing system to execute a method, the instructions comprising:

instructions configured to determine that the multi-screen device has been rotated from a first open state to a second different open state;

instructions configured to determine a relative position of a first screen that originally displayed data from a first application when the multi-screen device was in the first open state and a second screen that was not displaying data from the first application when the multi-screen device was in the first open state; and

instructions configured to control data displayed on the second screen in the second open state based on the determined relative position of the first screen and second screen.

11. The computer-readable medium of claim 10, wherein the first screen corresponds to a primary screen and wherein the second screen corresponds to a secondary screen.

12. The method of claim 10, wherein the first open state corresponds to an open portrait state and wherein the second open state corresponds to an open landscape state in which the first screen is positioned above the second screen.

13. The computer-readable medium of claim 12, wherein the first application is automatically maximized to extend across the first and second screens afrer the device enters the second state.

14. The computer-readable medium of claim 13, wherein the data from the first application is automatically displayed on both the first and second screens in at least one of a dual landscape output configuration and a landscape max output configuration and wherein maximization display rules are referenced to determine whether the first application is displayed in the landscape dual output configuration or the landscape max output configuration.

15. The computer-readable medium of claim 12, wherein data from the first application is only displayed on the first screen in the first state and wherein data from the first application is automatically displayed on both the first and second screens when the device enters the second state.

16. A multi-screen user device, comprising:

a first screen including a first display area;

a second screen including a second display area;

a hinge configured to connect the first screen with the second screen;

a sensor configured to produce an electrical signal indicative of a relative position of the first screen and the second screen; and

a computer-readable medium having instructions stored thereon that include:

a first set of instructions configured to determine that the multi-screen device has been rotated from a first open state to a second different open state;

a second set of instructions configured to determine a relative position of a first screen that originally displayed data from a first application when the multi-screen device was in the first open state and a second screen that was not displaying data from the first application when the multi-screen device was in the first open state; and

a third set of instructions configured to control data displayed on the second screen in the second open state based on the determined relative position of the first screen and second screen.

17. The device of claim 16, wherein the first application is a multi-screen application and wherein the third set of instructions configured to automatically control data displayed on the first and second displays is further configured to automatically maximize the first application in response to receiving an indication that the device has been rotated to the second open state.

18. The device of claim 17, wherein the first open state corresponds to an open portrait state and wherein the second open state corresponds to an open landscape state.

19. The device of claim 17, wherein the data from the first application is automatically displayed on both the first and second displays in a landscape max output configuration.

20. The device of claim 17, wherein the data from the first application is automatically displayed on both the first and second displays in a landscape dual output configuration.

ABSTRACT

A multi-screen user device and methods for controlling data displayed thereby are disclosed. Specifically, the data displayed by the multiple screens of the multi-screen user device is conditioned upon the relative position of the multiple screens. A gravity-drop display feature is also disclosed in which data from a first application on a first screen is automatically displayed on a second screen when the device is rotated.

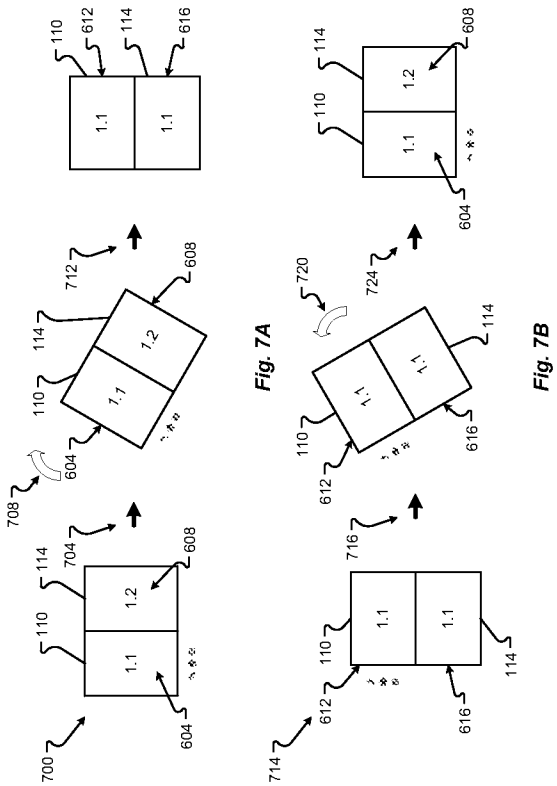


Fig. 7A

Fig. 7B

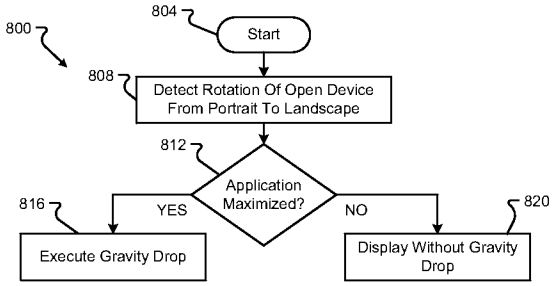


Fig. 8

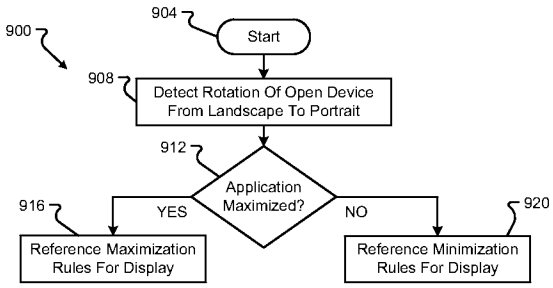


Fig. 9

DUAL-SCREEN VIEW IN RESPONSE TO ROTATION

[0001] Display Controls:

[0002] Fig. 7 depicts one example of gesture sequence 700 that can cause the device 100 to alter the display of data thereon. Although the sequence and display actions are depicted as being performed on the touch sensitive displays 110, 114, it should be appreciated that the first display action can alternatively, or additionally, be performed on other parts of the primary screen 104 and/or secondary screen 108. The first gesture sequence 700 begins when the device 100 is in an open portrait state 320. While in this state, the device 100 displays data on both its displays 110, 114. In some embodiments, the device 100 may display application data from different applications (e.g., a first and second application 1.1 and 1.2, respectively) on the displays 110, 114. In the depicted example, a first set of data from a first application 1.1 may initially be displayed in a first portrait configuration 604 and a second set of data from a second application 1.2 may be initially displayed in a second portrait configuration 608. In other words, the first and second displays 110, 114 may depict data from different application in either a PD output configuration. It should be appreciated, however, that one of the displays may be depicting a desktop rather than data from an application. Also, it may be possible that both of the displays are depicting data from a single application (e.g., the first application 1.1).

[0003] Fig. 7 also depicts a simplified set of applications 708 that generally indicates the order in which applications are configured in an application stack. In some embodiments, some of the applications from the application stack 708 may be active applications 712 and other applications may be inactive 716 or in standby. The active applications 712 may correspond to those applications in the application stack 708 that are currently being displayed by the device 100. The other applications (e.g., the inactive applications 716) may correspond to applications that have been opened by a user and are running in the background, but are not necessarily being rendered for display by the device 100.

[0004] In some embodiments, a single display stack may be used to determine which applications are displayed on the screens 104, 108 of the device 100. In other embodiments, each screen 104, 108 may have its own dedicated display stack and each display stack may have its own set of active 712 and inactive 716 applications. The manner in which data is displayed from the stack and the way in which the stack is managed can vary without departing from the

scope of the present disclosure. The purpose of displaying an example display stack 708 is to help show how different applications can be displayed as the gesture sequence 700 progresses.

[0005] The gesture sequence 700 continues when a first user input 704 is received that causes the device 100 to rotate. In some embodiments, the device 100 may be rotated in a counter-clockwise direction by a predetermined amount (e.g., 90 degrees), although such a rotation is not required. Rather, a clockwise rotation by 90 degrees or any rotation by a predetermined amount greater than or less than 90 degrees may also qualify as the first user input 704. The rotation of the device 100 may be detected, in some embodiments, by registering a rotation event with the task management module 540 which subsequently informs the MDM module 524 of the rotation. In some embodiments, the rotation event is registered by receiving one or more inputs from the accelerometer 176. During the rotation, the displays 110, 114 may continue to display the same data as before the first input 704 was received.

[0006] After the rotation event has ended, the device 100 may be oriented in an open landscape state 348. At this point, the MDM module 524 may determine the relative position of the displays 110, 114 (e.g., determine which display or screen 104, 108 is above the other) and based on the relative position of the displays in the final orientation, the MDM module 524 may instruct the window management module 532 to cause data from the application in the upper display (display 114 in this example) to "drop" down onto the lower display. In this example, data from the second application 1.2 may be automatically maximized across both displays 110, 114. Specifically, the data from the application that is displayed on the upper display 114 is the application which has its data automatically dropped onto the lower display 110. This automated dropping of the data from the upper display may be referred to as a gravity drop of the data displayed because the data is "dropped" under the force of gravity as detected by the accelerometer 176.

[0007] As discussed above, the data originally displayed by the display that eventually becomes the upper display may be obtained from an application. In some embodiments, the data from the second application 1.2 may be automatically maximized into an Lmax output configuration 628. Accordingly, the number of active applications 712 in the application stack 708 may automatically change from two applications (or more) to one application. Since the second application 1.2 is now being displayed on both displays 110, 114, the other applications in the application stack 708 may be considered inactive applications 716 as indicated by the

dotted lines. The inactive applications 716 may still be running or they may be in a paused or sleeping state such that no additional processing is performed for/by the applications until they become active applications 712.

[0008] The gesture sequence 700 continues when a second user input 724 is received that again causes the device 100 to rotate. In some embodiments, the second user input 724 may correspond to an input that is opposite to the first user input 704 (e.g., an input which reverses and undoes the first user input 704). Accordingly, if the first user input 704 corresponds to a counter-clockwise rotation of the device 100 by 90 degrees, then the second user input 724 may correspond to a clockwise rotation of the device 100 by 90 degrees. Of course, the second user input 724 may correspond to any other rotational input such as a rotation of the device 100 in the same direction as the first user input 704 by 90 degrees or 270 degrees. In any event, the second user input 724 may correspond to any input or combination of inputs that causes the device 100 to end up in an open portrait state 320.

[0009] The rotation of the device 100 may be detected, in some embodiments, by registering a rotation event with the task management module 540 which subsequently informs the MDM module 524 of the rotation. In some embodiments, the rotation event is registered by receiving one or more inputs from the accelerometer 176. During the rotation, the displays 110, 114 may continue to display the same data as before the second input 724 was received.

[0010] After the rotation event has ended, the device 100 may be oriented in an open portrait state 320. At this point, the MDM module 524 may determine the device 100 has been oriented such that the touch sensitive display 110 of the primary screen 104 is on the left and the touch sensitive display 114 of the secondary screen 108 is on the right. The MDM module 524 may provide this orientation information to the window management module 532 which re-configures the data presented by the displays 110, 114. Specifically, the window management module 532 may cause the data from the second application 1.2 (e.g., the application which was previously being displayed in an Lmax output configuration 628) to be displayed on the display of the left screen (e.g. the primary screen 104 in the depicted example). This means that the other display 114 is free to depict data from a different application. In some embodiments, the window management module 532 may select data from the next application in the display stack 708 for the display 114. This may correspond to a third application 1.3, at which point the second and third applications 1.2 and 1.3, respectively, become active applications 712 and the other

applications in the application display stack 708 are inactive applications 716. Data from the second and third applications 1.2, 1.3 may be displayed in a PD output configuration. It should be appreciated that more than one application may be displayed on either or both displays 110, 114 depending upon the stacking relationship of the applications and the display rules referenced by the window management module 532.

[0011] At this point the active applications 712 have changed from the first and second applications 1.1 and 1.2, respectively, to the second and third applications 1.2 and 1.3, respectively. This shifting in active applications 712 is caused by the rotation and reversed rotation (e.g., first user input 704 and second user input 724). Accordingly, the combination of user inputs 704, 724 may correspond to a series of user inputs that cause the device 100 to shift the application data that is displayed on the displays 110, 114. Specifically, the combination of user inputs 704, 724 may correspond to a series of user inputs that toggle active applications 712 in the display stack 708.

[0012] Referring now to Fig. 8, a display control method 800 will be described in accordance with at least some embodiments of the present disclosure. The method 800 is initiated at step 804. Specifically, a device 100 may be initially in an open portrait state 320. In this state, data from a first application 1.1 may be displayed on a first display 110 and data from a second application 1.2 may be displayed on a second display 114. The data from the applications may be displayed in a PD output configuration.

[0013] The method continues when the rotation of the device 100 is detected (step 808). Specifically, the transition of the device 100 from an open portrait state 320 to an open landscape state 348 may be detected in step 808.

[0014] The method continues with the MDM module 524 and/or window management module 532 storing information about the display stack 708 and specifically which application(s) were active application(s) prior to rotation and which application(s) were inactive application(s) 716 prior to rotation (step 812). The MDM module 524 and/or window management module 532 may also store the information regarding active/inactive applications prior to rotation.

[0015] Thereafter, or simultaneously, the MDM module 524 and/or window management module 532 may cause data from one of the originally-presented applications to be automatically depicted across both displays 110, 114 (step 816). Specifically, the window management module 532 may reference display rules to decide how to automatically re-render the display of

the application in the open landscape state 348. Even more specifically, the window management module 532 may execute a gravity drop and automatically maximize data from the upper display across both displays in an Lmax output configuration 628.

[0016] Thereafter, the MDM module 524 will wait until another rotation event is detected (step 820). Once another rotation event is detected (e.g. corresponding to the second user input 724), the MDM module 524 may instruct the window management module 532 that the data from the application previously maximized across both displays 110, 114 in the Lmax output configuration 628 is to be displayed on the display 110 of the primary screen 104 (step 824). The window management module 532 knows which applications are active and inactive because such information was previously stored in step 812.

[0017] Simultaneous with or afrer step 824, the window management module 532 may cause data from the next application in the display stack 708 to be displayed on the other screen of the device 100 (step 828). In some embodiments, the next application which ultimately has some of its data displayed in step 828 may correspond to an application that was inactive in step 808 and step 812.

[0018] The exemplary systems and methods of this disclosure have been described in relation to display control mechanisms. However, to avoid unnecessarily obscuring the present disclosure, the preceding description omits a number of known structures and devices. This omission is not to be construed as a limitation of the scopes of the claims. Specific details are set forth to provide an understanding of the present disclosure. It should however be appreciated that the present disclosure may be practiced in a variety of ways beyond the specific detail set forth herein.

[0019] Furthermore, while the exemplary aspects, embodiments, and/or configurations illustrated herein show the various components of the system collocated, certain components of the system can be located remotely, at distant portions of a distributed network, such as a LAN and/or the Internet, or within a dedicated system. Thus, it should be appreciated, that the components of the system can be combined in to one or more devices, such as a computer, laptop, netbook, smart phone, mobile device, etc., or collocated on a particular node of a distributed network, such as an analog and/or digital telecommunications network, a packet-switch network, or a circuit-switched network. It will be appreciated from the preceding description, and for reasons of computational efficiency, that the components of the system can

be arranged at any location within a distributed network of components without affecting the operation of the system. For example, the various components can be located in a switch such as a PBX and media server, gateway, in one or more communications devices, at one or more users' premises, or some combination thereof. Similarly, one or more functional portions of the system could be distributed between a telecommunications device(s) and an associated computing device.

[0020] Furthermore, it should be appreciated that the various links connecting the elements can be wired or wireless links, or any combination thereof, or any other known or later developed element(s) that is capable of supplying and/or communicating data to and from the connected elements. These wired or wireless links can also be secure links and may be capable of communicating encrypted information. Transmission media used as links, for example, can be any suitable carrier for electrical signals, including coaxial cables, copper wire and fiber optics, and may take the form of acoustic or light waves, such as those generated during radio-wave and infra-red data communications.

[0021] Also, while the flowcharts have been discussed and illustrated in relation to a particular sequence of events, it should be appreciated that changes, additions, and omissions to this sequence can occur without materially affecting the operation of the disclosed embodiments, configuration, and aspects.

[0022] A number of variations and modifications of the disclosure can be used. It would be possible to provide for some features of the disclosure without providing others.

[0023] In yet another embodiment, the systems and methods of this disclosure can be implemented in conjunction with a special purpose computer, a programmed microprocessor or microcontroller and peripheral integrated circuit element(s), an ASIC or other integrated circuit, a digital signal processor, a hard-wired electronic or logic circuit such as discrete element circuit, a programmable logic device or gate array such as PLD, PLA, FPGA, PAL, special purpose computer, any comparable means, or the like. In general, any device(s) or means capable of implementing the methodology illustrated herein can be used to implement the various aspects of this disclosure. Exemplary hardware that can be used for the disclosed embodiments, configurations and aspects includes computers, handheld devices, telephones (e.g., cellular, Internet enabled, digital, analog, hybrids, and others), and other hardware known in the art. Some of these devices include processors (e.g., a single or multiple microprocessors), memory,

nonvolatile storage, input devices, and output devices. Furthermore, alternative software implementations including, but not limited to, distributed processing or component/object distributed processing, parallel processing, or virtual machine processing can also be constructed to implement the methods described herein.

[0024] In yet another embodiment, the disclosed methods may be readily implemented in conjunction with software using object or object-oriented software development environments that provide portable source code that can be used on a variety of computer or workstation platforms. Alternatively, the disclosed system may be implemented partially or fully in hardware using standard logic circuits or VLSI design. Whether software or hardware is used to implement the systems in accordance with this disclosure is dependent on the speed and/or efficiency requirements of the system, the particular function, and the particular software or hardware systems or microprocessor or microcomputer systems being utilized.

[0025] In yet another embodiment, the disclosed methods may be partially implemented in software that can be stored on a storage medium, executed on programmed general-purpose computer with the cooperation of a controller and memory, a special purpose computer, a microprocessor, or the like. In these instances, the systems and methods of this disclosure can be implemented as program embedded on personal computer such as an applet, JAVA® or CGI script, as a resource residing on a server or computer workstation, as a routine embedded in a dedicated measurement system, system component, or the like. The system can also be implemented by physically incorporating the system and/or method into a software and/or hardware system.

[0026] Although the present disclosure describes components and functions implemented in the aspects, embodiments, and/or configurations with reference to particular standards and protocols, the aspects, embodiments, and/or configurations are not limited to such standards and protocols. Other similar standards and protocols not mentioned herein are in existence and are considered to be included in the present disclosure. Moreover, the standards and protocols mentioned herein and other similar standards and protocols not mentioned herein are periodically superseded by faster or more effective equivalents having essentially the same functions. Such replacement standards and protocols having the same functions are considered equivalents included in the present disclosure.

[0027] The present disclosure, in various aspects, embodiments, and/or configurations, includes components, methods, processes, systems and/or apparatus substantially as depicted and described herein, including various aspects, embodiments, configurations embodiments, subcombinations, and/or subsets thereof. Those of skill in the art will understand how to make and use the disclosed aspects, embodiments, and/or configurations afrer understanding the present disclosure. The present disclosure, in various aspects, embodiments, and/or configurations, includes providing devices and processes in the absence of items not depicted and/or described herein or in various aspects, embodiments, and/or configurations hereof, including in the absence of such items as may have been used in previous devices or processes, e.g., for improving performance, achieving ease and\or reducing cost of implementation.

[0028] The foregoing discussion has been presented for purposes of illustration and description. The foregoing is not intended to limit the disclosure to the form or forms disclosed herein. In the foregoing Detailed Description for example, various features of the disclosure are grouped together in one or more aspects, embodiments, and/or configurations for the purpose of streamlining the disclosure. The features of the aspects, embodiments, and/or configurations of the disclosure may be combined in alternate aspects, embodiments, and/or configurations other than those discussed above. This method of disclosure is not to be interpreted as reflecting an intention that the claims require more features than are expressly recited in each claim. Rather, as the following claims reflect, inventive aspects lie in less than all features of a single foregoing disclosed aspect, embodiment, and/or configuration. Thus, the following claims are hereby incorporated into this Detailed Description, with each claim standing on its own as a separate preferred embodiment of the disclosure.

[0029] Moreover, though the description has included description of one or more aspects, embodiments, and/or configurations and certain variations and modifications, other variations, combinations, and modifications are within the scope of the disclosure, e.g., as may be within the skill and knowledge of those in the art, afrer understanding the present disclosure. It is intended to obtain rights which include alternative aspects, embodiments, and/or configurations to the extent permitted, including alternate, interchangeable and/or equivalent structures, functions, ranges or steps to those claimed, whether or not such alternate, interchangeable and/or equivalent structures, functions, ranges or steps are disclosed herein, and without intending to publicly dedicate any patentable subject matter.

What is claimed is:

1. A method of controlling data displayed by a multi-screen device, comprising:

determining that the multi-screen device has been rotated a first time from a first open state to a second different open state, wherein data from a first and second application are displayed while the device is in the first open state and wherein data from the first application is not displayed while the device is in the second open state;

determining that the device has been rotated a second time afrer the first time; and

in response to determining that the device has been rotated the second time, automatically causing data from the second application and a third application to be displayed on the device.

2. The method of claim 1, wherein the first open state corresponds to an open portrait state and wherein the second open state corresponds to an open landscape state in which a first screen of the device is positioned above a second screen of the device, wherein the data from the first application is displayed on the first screen prior to the rotation of the device, and wherein the data from the second application is displayed on the second screen prior to the rotation of the device.

3. The method of claim 2, wherein the data from the second application is displayed on the first screen afrer the rotations of the device, wherein the data from the third application is displayed on the second screen afrer the rotations of the device, wherein the data from the third application is not displayed prior to the rotation of the device, and wherein data from the first application is not displayed after the rotations of the device.

4. The method of claim 2, wherein the rotation of the device is determined, at least in part, based on data received from an accelerometer.

5. The method of claim 2, wherein the device is in the open portrait state after the device is rotated the second time.

6. The method of claim 5, wherein the data from the second device is automatically displayed on both the first and second screens while the device is in the second open state.

7. The method of claim 6, wherein data from the second application is displayed in a landscape max output configuration.

8. The method of claim 1, wherein the third application is different from the first application.

9. A non-transitory computer-readable medium having stored thereon instructions that cause a computing system to execute a method, the instructions comprising:

instructions configured to determine that a multi-screen device has been rotated a first time from a first open state to a second different open state, wherein data from a first and second application are displayed while the device is in the first open state and wherein data from the first application is not displayed while the device is in the second open state;

instructions configured to determine that the device has been rotated a second time afrer the first time; and

instructions configured to automatically cause data from the second application and a third application to be displayed on the device in response to determining that the device has been rotated the second time.

10. The computer-readable medium of claim 9, wherein the first open state corresponds to an open portrait state and wherein the second open state corresponds to an open landscape state in which a first screen of the device is positioned above a second screen of the device, wherein the data from the first application is displayed on the first screen prior to the rotation of the device, and wherein the data from the second application is displayed on the second screen prior to the rotation of the device.

11. The method of claim 10, wherein the data from the second application is displayed on the first screen afrer the rotations of the device, wherein the data from the third application is displayed on the second screen afrer the rotations of the device, wherein the data from the third application is not displayed prior to the rotation of the device, and wherein data from the first application is not displayed afrer the rotations of the device.

12. The computer-readable medium of claim 10, wherein the rotation of the device is determined, at least in part, based on data received from an accelerometer.

13. The computer-readable medium of claim 10, wherein the device is in the open portrait state afrer the device is rotated the second time.

14. The computer-readable medium of claim 13, wherein the data from the second device is automatically displayed on both the first and second screens while the device is in the second open state and wherein data from the second application is displayed in a landscape max output configuration.

15. A multi-screen user device, comprising:

a first screen including a first display area;

a second screen including a second display area;

a sensor configured to produce an electrical signal indicative of a rotation of the user device; and

a computer-readable medium having instructions stored thereon that include:

a first set of instructions configured to determine that a multi-screen device has been rotated a first time from a first open state to a second different open state, wherein data from a first and second application are displayed while the device is in the first open state and wherein data from the first application is not displayed while the device is in the second open state;

a second set of instructions configured to determine that the device has been rotated a second time afrer the first time; and

a third set of instructions configured to automatically cause data from the second application and a third application to be displayed on the device in response to determining that the device has been rotated the second time.

16. The device of claim 15, wherein the first open state corresponds to an open portrait state and wherein the second open state corresponds to an open landscape state in which a first screen of the device is positioned above a second screen of the device.

17. The device of claim 16, wherein the data from the first application is displayed on the first screen prior to the rotation of the device and wherein the data from the second application is displayed on the second screen prior to the rotation of the device.

18. The device of claim 17, wherein the data from the second application is displayed on the first screen afrer the rotations of the device, wherein the data from the third application is displayed on the second screen afrer the rotations of the device, wherein the data from the third application is not displayed prior to the rotation of the device, and wherein data from the first application is not displayed after the rotations of the device.

19. The device of claim 18, wherein the device is in the open portrait state afrer the device is rotated the second time

20. The device of claim 16, wherein the sensor comprises at least one accelerometer.

ABSTRACT

A multi-screen user device and methods for controlling data displayed thereby are disclosed. Specifically, a gesture sequence is disclosed which enables a user to toggle or shift though applications that are displayed by the multi-screen user device. The gesture sequence may correspond to various rotation or partial rotations of the multi-screen user device.

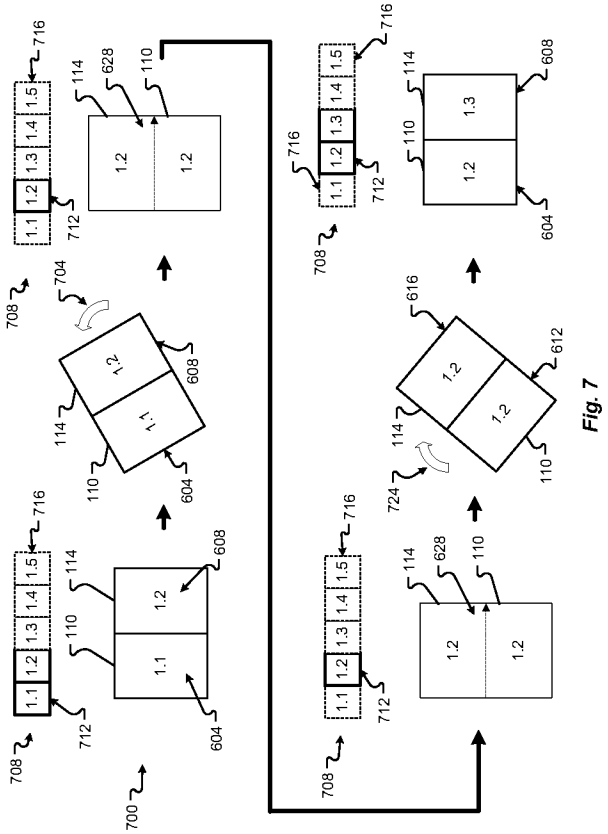

716
1.5
1.4
1.3
1.2
1.1
712
114
628
110
1.2
1.2
608
604
704
708
700
616
612
724

Fig. 7

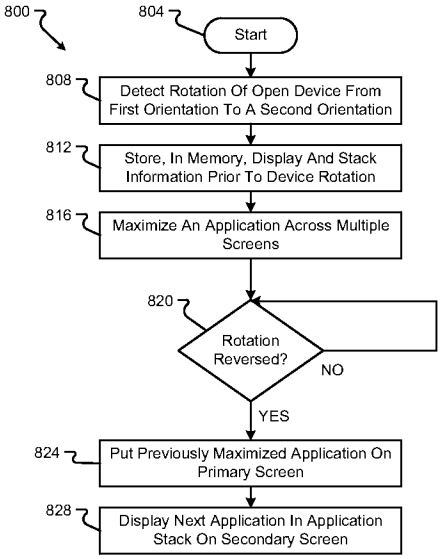

800
804
Start
808
Detect Rotation Of Open Device From First Orientation To A Second Orientation
812
Store, In Memory, Display And Stack Information Prior To Device Rotation
816
Maximize An Application Across Multiple Screens
820
Rotation Reversed?
NO
YES
824
Put Previously Maximized Application On Primary Screen
828
Display Next Application In Application Stack On Secondary Screen

Fig. 8

SINGLE-SCREEN VIEW IN RESPONSE TO ROTATION

[0001] Display Controls:

[0002] Figs. 7A-D depict several examples of gesture sequences that can cause the device 100 to alter the display of data thereon. Although the sequences and display actions are depicted as being performed on the touch sensitive displays 110, 114, it should be appreciated that any of the display actions can alternatively, or additionally, be performed on other parts of the primary screen 104 and/or secondary screen 108.

[0003] Referring initially to Fig. 7A, a first gesture sequence 700 begins when the device 100 is in an open portrait state 320. While in this state, the device 100 displays data on both its displays 110, 114. In some embodiments, the device 100 may display application data from a first application 1.1 on one of its displays 110 and data from a first desktop D1 on the other of its displays 114. In the depicted example, a first set of data from a first application 1.1 may initially be displayed in a first portrait configuration 604 and a second set of data from a first desktop D1 may be initially displayed in a second portrait configuration 608. In other words, the first and second displays 110, 114 may depict data from both an application and a desktop in a PD output configuration. It should be appreciated, however, that both of the displays may be depicting data from an application rather than the first desktop D1. In some embodiments, data from the first application 1.1 is in a minimized state and, therefore, is only displayed on one of the displays 110.

[0004] The gesture sequence 700 continues when a first user input 704 is received that causes the device 100 to rotate. In some embodiments, the device 100 may be rotated in a clockwise direction by a predetermined amount (e.g., 90 degrees), although such a rotation is not required. Rather, a counter-clockwise rotation by 90 degrees or any rotation by a predetermined amount greater than or less than 90 degrees may also qualify as the first user input 704. The rotation of the device 100 may be detected, in some embodiments, by registering a rotation event with the task management module 540 which subsequently informs the MDM module 524 of the rotation. In some embodiments, the rotation event is registered by receiving one or more inputs from the accelerometer 176. During the rotation, the displays 110, 114 may continue to display the same data as before the first input 704 was received.

[0005] After the rotation event has ended, the device 100 may be oriented in an open landscape state 348. At this point, the MDM module 524 may determine the relative position of the displays 110, 114 (e.g., determine which display or screen 104, 108 is above the other) and based on the relative position of the displays in the final orientation, the MDM module 524 may instruct the window management module 532 to cause data from the first application 1.1 in the upper display (display 110 in this example) to "drop" down onto the lower display (display 114 in this example). In this example, data from the first application 1.1 may be automatically maximized across both displays 110, 114. Specifically, the data from the application that is displayed on the upper display 110 is the application which has its data automatically dropped onto the lower display 114. This automated dropping of the data from the upper display may be referred to as a gravity drop of the data displayed because the data is "dropped" under the force of gravity as detected by the accelerometer 176.

[0006] As discussed above, the data originally displayed by the display that eventually becomes the upper display 110 may be obtained from the first application 1.1. In some embodiments, the data from the first application 1.1 may be automatically maximized into an Lmax output configuration 628. It may also be possible that data from the first application 1.1 is displayed in an LD output configuration.

[0007] Fig. 7B depicts a second gesture sequence 708, which may be a continuation of the first gesture sequence 700 or a stand-alone sequence. Where the second gesture sequence 708 is a continuation of the first gesture sequence 700, the device 100 is rotated back from the open landscape state 348. A second user input 712 is received which results in the device 100 being rotated back to the open portrait state 320. In some embodiments, the second user input 712 may correspond to an input that is opposite to the first user input 704 (e.g., an input which reverses and undoes the first user input 704). Accordingly, if the first user input 704 corresponds to a clockwise rotation of the device 100 by 90 degrees, then the second user input 712 may correspond to a counter-clockwise rotation of the device 100 by 90 degrees. Of course, the second user input 712 may correspond to any other rotational input such as a rotation of the device 100 in the same direction as the first user input 704 by 90 degrees or 270 degrees. In any event, the second user input 712 may correspond to any input or combination of inputs that causes the device 100 to end up in an open portrait state 320.

[0008] When rotating back to the open portrait state 320, the original state of the first application 1.1 (e.g., the display state of the first application 1.1 prior to receiving the first user input 704) is retained. This means that the first application 1.1 is now minimized back to the

single display 110. This also means that the first desktop D1 may be displayed on the other display 114. The first application 1.1 may be displayed in a first portrait configuration 604 and the first desktop D1 may be displayed in a second portrait configuration 608.

[0009] It should be noted, however, if any manual minimization of the first application 1.1 or navigation in the gesture capture regions 120, 124 is detected between the first user input 704 and second user input 712, then the original state of the first application 1.1 need not be maintained (although it may). For example, minimizing the first application 1.1 to the bottom display 114 while the device is in the open landscape state 348 and then rotating the device 100 back to the open portrait state 320 would result in the first application 1.1 being displayed according to minimization display rules on the second display 114 (and not the first display 110). If this manual minimization occurred, then the first desktop D1 or some other application may be displayed on the first display 110 in lieu of the first application 1.1.

[0010] The rotation inputs 704 and/or 712 of the device 100 may be detected, in some embodiments, by registering a rotation event with the task management module 540 which subsequently informs the MDM module 524 of the rotation. In some embodiments, the rotation event is registered by receiving one or more inputs from the accelerometer 176. During the rotation, the displays 110, 114 may continue to display the same data as before the inputs 704 and/or 712 were received.

[0011] Referring now to Fig. 7C, a third gesture sequence 716 begins when the device 100 is in an open portrait state 320. While in this state, the device 100 displays data on both its displays 110, 114. In some embodiments, the device 100 may display application data from a first application 1.1 on one of its displays 114 and data from a first desktop D1 on the other of its displays 110. In the depicted example, a first set of data from a first application 1.1 may initially be displayed in a second portrait configuration 608 and a second set of data from a first desktop D1 may be initially displayed in a first portrait configuration 604. In other words, the first and second displays 110, 114 may depict data from both an application and a desktop in a PD output configuration. It should be appreciated, however, that both of the displays may be depicting data from an application rather than the first desktop D1. In some embodiments, data from the first application 1.1 is in a minimized state and, therefore, is only displayed on one of the displays 110.

[0012] The gesture sequence 716 continues when a third user input 720 is received that causes the device 100 to rotate. In some embodiments, the third user input 720 is identical but opposite the first user input 704. In some embodiments, the device 100 may be rotated in a counter-clockwise direction by a predetermined amount (e.g., 90 degrees), although such a rotation is not required. Rather, a clockwise rotation by 90 degrees or any rotation by a predetermined amount greater than or less than 90 degrees may also qualify as the third user input 720. The rotation of the device 100 may be detected, in some embodiments, by registering a rotation event with the task management module 540 which subsequently informs the MDM module 524 of the rotation. In some embodiments, the rotation event is registered by receiving one or more inputs from the accelerometer 176. During the rotation, the displays 110, 114 may continue to display the same data as before the third input 720 was received.

[0013] After the rotation event has ended, the device 100 may be oriented in an open landscape state 348. At this point, the MDM module 524 may determine the relative position of the displays 110, 114 (e.g., determine which display or screen 104, 108 is above the other) and based on the relative position of the displays in the final orientation, the MDM module 524 may instruct the window management module 532 to cause data from the first application 1.1 in the upper display (display 114 in this example) to "drop" down onto the lower display (display 110 in this example). In this example, data from the first application 1.1 may be automatically maximized across both displays 110, 114. Specifically, the data from the application that is displayed on the upper display 114 is the application which has its data automatically dropped onto the lower display 110. This automated dropping of the data from the upper display may be referred to as a gravity drop of the data displayed because the data is "dropped" under the force of gravity as detected by the accelerometer 176.

[0014] As discussed above, the data originally displayed by the display that eventually becomes the upper display 114 may be obtained from the first application 1.1. In some embodiments, the data from the first application 1.1 may be automatically maximized into an Lmax output configuration 628. It may also be possible that data from the first application 1.1 is displayed in an LD output configuration.

[0015] Fig. 7D depicts a fourth gesture sequence 724, which may be a continuation of the third gesture sequence 716 or a stand-alone sequence. Where the fourth gesture sequence 724 is a continuation of the third gesture sequence 716, the device 100 is rotated back from the open

landscape state 348. A fourth user input 728 is received which results in the device 100 being rotated back to the open portrait state 320. In some embodiments, the fourth user input 728 may correspond to an input that is opposite to the third user input 720 (e.g., an input which reverses and undoes the third user input 720). Accordingly, if the third user input 720 corresponds to a clockwise rotation of the device 100 by 90 degrees, then the fourth user input 728 may correspond to a clockwise rotation of the device 100 by 90 degrees. Of course, the fourth user input 728 may correspond to any other rotational input such as a rotation of the device 100 in the same direction as the third user input 720 by 90 degrees or 270 degrees. In any event, the fourth user input 728 may correspond to any input or combination of inputs that causes the device 100 to end up in an open portrait state 320.

[0016] When rotating back to the open portrait state 320, the original state of the first application 1.1 (e.g., the display state of the first application 1.1 prior to receiving the third user input 720) is retained. This means that the first application 1.1 is now minimized back to the single display 114. This also means that the first desktop D1 may be displayed on the other display 110. The first application 1.1 may be displayed in a second portrait configuration 608 and the first desktop D1 may be displayed in a first portrait configuration 604.

[0017] It should be noted, however, if any manual minimization of the first application 1.1 or navigation in the gesture capture regions 120, 124 is detected between the third user input 720 and fourth user input 728, then the original state of the first application 1.1 need not be maintained (although it may). For example, minimizing the first application 1.1 to the bottom display 110 while the device is in the open landscape state 348 and then rotating the device 100 back to the open portrait state 320 would result in the first application 1.1 being displayed according to minimization display rules on the first display 114 (and not the second display 114). If this manual minimization occurred, then the first desktop D1 or some other application may be displayed on the second display 114 in lieu of the first application 1.1.

[0018] The rotation inputs 720 and/or 728 of the device 100 may be detected, in some embodiments, by registering a rotation event with the task management module 540 which subsequently informs the MDM module 524 of the rotation. In some embodiments, the rotation event is registered by receiving one or more inputs from the accelerometer 176. During the rotation, the displays 110, 114 may continue to display the same data as before the inputs 720 and/or 728 were received.

[0019] Referring now to Fig. 8, a display control method 800 will be described in accordance with at least some embodiments of the present disclosure. The method 800 is initiated at step 804. Specifically, a device 100 may be initially in an open portrait state 320. In this state, data from a first application 1.1 may be displayed on one display of the device 100 and data from something other than the first application 1.1 (e.g., another application or a desktop) may be displayed on the other display. The data from the application and the other data may be displayed in a PD output configuration.

[0020] The method continues when the rotation of the device 100 is detected (step 808). Specifically, the transition of the device 100 from an open portrait state 320 to an open landscape state 348 may be detected in step 808.

[0021] The method continues with the MDM module 524 and/or window management module 532 storing information about the display characteristics of the first application prior to receiving the rotation input (step 812). The MDM module 524 and/or window management module 532 may also store the information regarding the data that was displayed on the other display of the device 100.

[0022] Thereafter, or simultaneously, the MDM module 524 and/or window management module 532 may cause data from the first application 1.1 to be automatically depicted across both displays 110, 114 (step 816). The manner in which the first application 1.1 is displayed may be controlled by landscape maximization output rules for the first application 1.1. All the while, the minimization output rules for the first application 1.1 are maintained. Specifically, the window management module 532 may reference display rules to decide how to automatically re-render the display of the application in the open landscape state 348. Even more specifically, the window management module 532 may execute a gravity drop and automatically maximize data from the upper display across both displays in an Lmax output configuration 628.

[0023] Thereafter, the MDM module 524 will check to determine if a stack-altering input has been received (step 820). A stack-altering input may correspond to any input which causes the device 100 to no longer save the display configurations of the first application 1.1 prior to receiving the rotation input. Specific examples of a stack-altering input include, without limitation, a manual instruction to minimize the first application 1.1 or a navigation input received at one or both of the gesture capture regions 120, 124.

[0024] If any stack-altering input is received, then the MDM module 524 and/or window management module 532 can delete the previously-stored display configurations of the first application 1.1 (step 824). Specifically, because a stack-altering input was received, the MDM module 524 and/or window management module 532 determines that the data stored in step 812 is no longer necessary and such data is discarded or ignored.

[0025] The device 100 will also wait to receive another rotation input (step 828). Once another rotation event is detected (e.g. corresponding to the second or fourth user input), the MDM module 524 may instruct the window management module 532 that the data from the first application needs to be re-rendered.

[0026] In particular, the window management module 532 may refer to currently-stored display orientation rules for the first application 1.1 and render a display of the first application 1.1 according to such rules (steps 832, 836). If a stack-altering input was received prior to the rotation input at step 828, then the window management module 532 will display the first application 1.1 according to default display configurations. If, however, no stack-altering input was received, then the window management module 532 will refer to the display configurations stored in step 812 to re-display the first application 1.1 in the same way that is was displayed prior to receiving the first rotation input.

[0027] The exemplary systems and methods of this disclosure have been described in relation to display control mechanisms. However, to avoid unnecessarily obscuring the present disclosure, the preceding description omits a number of known structures and devices. This omission is not to be construed as a limitation of the scopes of the claims. Specific details are set forth to provide an understanding of the present disclosure. It should however be appreciated that the present disclosure may be practiced in a variety of ways beyond the specific detail set forth herein.

[0028] Furthermore, while the exemplary aspects, embodiments, and/or configurations illustrated herein show the various components of the system collocated, certain components of the system can be located remotely, at distant portions of a distributed network, such as a LAN and/or the Internet, or within a dedicated system. Thus, it should be appreciated, that the components of the system can be combined in to one or more devices, such as a computer, laptop, netbook, smart phone, mobile device, etc., or collocated on a particular node of a distributed network, such as an analog and/or digital telecommunications network, a packet-

switch network, or a circuit-switched network. It will be appreciated from the preceding description, and for reasons of computational efficiency, that the components of the system can be arranged at any location within a distributed network of components without affecting the operation of the system. For example, the various components can be located in a switch such as a PBX and media server, gateway, in one or more communications devices, at one or more users' premises, or some combination thereof. Similarly, one or more functional portions of the system could be distributed between a telecommunications device(s) and an associated computing device.

[0029] Furthermore, it should be appreciated that the various links connecting the elements can be wired or wireless links, or any combination thereof, or any other known or later developed element(s) that is capable of supplying and/or communicating data to and from the connected elements. These wired or wireless links can also be secure links and may be capable of communicating encrypted information. Transmission media used as links, for example, can be any suitable carrier for electrical signals, including coaxial cables, copper wire and fiber optics, and may take the form of acoustic or light waves, such as those generated during radio-wave and infra-red data communications.

[0030] Also, while the flowcharts have been discussed and illustrated in relation to a particular sequence of events, it should be appreciated that changes, additions, and omissions to this sequence can occur without materially affecting the operation of the disclosed embodiments, configuration, and aspects.

[0031] A number of variations and modifications of the disclosure can be used. It would be possible to provide for some features of the disclosure without providing others.

[0032] In yet another embodiment, the systems and methods of this disclosure can be implemented in conjunction with a special purpose computer, a programmed microprocessor or microcontroller and peripheral integrated circuit element(s), an ASIC or other integrated circuit, a digital signal processor, a hard-wired electronic or logic circuit such as discrete element circuit, a programmable logic device or gate array such as PLD, PLA, FPGA, PAL, special purpose computer, any comparable means, or the like. In general, any device(s) or means capable of implementing the methodology illustrated herein can be used to implement the various aspects of this disclosure. Exemplary hardware that can be used for the disclosed embodiments, configurations and aspects includes computers, handheld devices, telephones (e.g., cellular,

Internet enabled, digital, analog, hybrids, and others), and other hardware known in the art. Some of these devices include processors (e.g., a single or multiple microprocessors), memory, nonvolatile storage, input devices, and output devices. Furthermore, alternative software implementations including, but not limited to, distributed processing or component/object distributed processing, parallel processing, or virtual machine processing can also be constructed to implement the methods described herein.

[0033] In yet another embodiment, the disclosed methods may be readily implemented in conjunction with software using object or object-oriented software development environments that provide portable source code that can be used on a variety of computer or workstation platforms. Alternatively, the disclosed system may be implemented partially or fully in hardware using standard logic circuits or VLSI design. Whether software or hardware is used to implement the systems in accordance with this disclosure is dependent on the speed and/or efficiency requirements of the system, the particular function, and the particular software or hardware systems or microprocessor or microcomputer systems being utilized.

[0034] In yet another embodiment, the disclosed methods may be partially implemented in software that can be stored on a storage medium, executed on programmed general-purpose computer with the cooperation of a controller and memory, a special purpose computer, a microprocessor, or the like. In these instances, the systems and methods of this disclosure can be implemented as program embedded on personal computer such as an applet, JAVA® or CGI script, as a resource residing on a server or computer workstation, as a routine embedded in a dedicated measurement system, system component, or the like. The system can also be implemented by physically incorporating the system and/or method into a software and/or hardware system.

[0035] Although the present disclosure describes components and functions implemented in the aspects, embodiments, and/or configurations with reference to particular standards and protocols, the aspects, embodiments, and/or configurations are not limited to such standards and protocols. Other similar standards and protocols not mentioned herein are in existence and are considered to be included in the present disclosure. Moreover, the standards and protocols mentioned herein and other similar standards and protocols not mentioned herein are periodically superseded by faster or more effective equivalents having essentially the same functions. Such

replacement standards and protocols having the same functions are considered equivalents included in the present disclosure.

[0036] The present disclosure, in various aspects, embodiments, and/or configurations, includes components, methods, processes, systems and/or apparatus substantially as depicted and described herein, including various aspects, embodiments, configurations embodiments, subcombinations, and/or subsets thereof. Those of skill in the art will understand how to make and use the disclosed aspects, embodiments, and/or configurations after understanding the present disclosure. The present disclosure, in various aspects, embodiments, and/or configurations, includes providing devices and processes in the absence of items not depicted and/or described herein or in various aspects, embodiments, and/or configurations hereof, including in the absence of such items as may have been used in previous devices or processes, e.g., for improving performance, achieving ease and\or reducing cost of implementation.

[0037] The foregoing discussion has been presented for purposes of illustration and description. The foregoing is not intended to limit the disclosure to the form or forms disclosed herein. In the foregoing Detailed Description for example, various features of the disclosure are grouped together in one or more aspects, embodiments, and/or configurations for the purpose of streamlining the disclosure. The features of the aspects, embodiments, and/or configurations of the disclosure may be combined in alternate aspects, embodiments, and/or configurations other than those discussed above. This method of disclosure is not to be interpreted as reflecting an intention that the claims require more features than are expressly recited in each claim. Rather, as the following claims reflect, inventive aspects lie in less than all features of a single foregoing disclosed aspect, embodiment, and/or configuration. Thus, the following claims are hereby incorporated into this Detailed Description, with each claim standing on its own as a separate preferred embodiment of the disclosure.

[0038] Moreover, though the description has included description of one or more aspects, embodiments, and/or configurations and certain variations and modifications, other variations, combinations, and modifications are within the scope of the disclosure, e.g., as may be within the skill and knowledge of those in the art, after understanding the present disclosure. It is intended to obtain rights which include alternative aspects, embodiments, and/or configurations to the extent permitted, including alternate, interchangeable and/or equivalent structures, functions, ranges or steps to those claimed, whether or not such alternate, interchangeable and/or equivalent

structures, functions, ranges or steps are disclosed herein, and without intending to publicly dedicate any patentable subject matter.

What is claimed is:

1. A method of controlling data displayed by a multi-screen device, comprising:

receiving a first rotation input, the first rotation input corresponding to an indication that the multi-screen device has been rotated a first time from a first open state to a different second open state, wherein data from a first application is displayed while the device is in the first open state on a first screen of the device and not on a second screen of the device, and wherein data from the first application is displayed on the first and second screens of the device in the second open state;

receiving a second rotation input, the second rotation input corresponding to an indication that the device has been rotated a second time after the first time;

after receiving the second rotation input, referencing display rules for the first application; and

displaying the first application on one of the first and second screens according to the display rules for the first application.

2. The method of claim 1, wherein the display rules for the first application are stored prior to receiving the second rotation input and correspond to a display configuration of the first application when the device is in the first open state prior to receiving the first rotation input.

3. The method of claim 2, wherein the display rules for the first application indicate that the data from the first application is presented substantially identically to the way in which the data from the first application is presented prior to receiving the first rotation input.

4. The method of claim 1, wherein at least one of the first and second rotation inputs are received from an accelerometer.

5. The method of claim 1, further comprising:

receiving a first stack-altering input after the first rotation input but before the second rotation input; and

updating the display rules for the first application to correspond to display rules that are different from display rules for the first application prior to receiving the first rotation input.

6. The method of claim 5, wherein the first stack-altering input corresponds to at least one of a minimization instruction for the first application and an application-navigation instruction.

7. The method of claim 1, wherein data from a first desktop is displayed on the second screen prior to receiving the first rotation input, wherein the data from the first desktop is not displayed afrer receiving the first rotation input but before receiving the second rotation input, and wherein the data from the first desktop is re-displayed on the second screen after receiving the second rotation input.

8. The method of claim 7, wherein the data from the first application is automatically maximized and displayed across both the first and second screens afrer receiving the first rotation input.

9. The method of claim 1, wherein the first rotation input corresponds to a rotation of the device by about 90 degrees in a first direction and wherein the second rotation input corresponds to a rotation of the device by about 90 degrees in a second direction that is opposite the first direction.

10. A non-transitory computer-readable medium having stored thereon instructions that cause a computing system to execute a method, the instructions comprising:

instructions configured to receive a first rotation input, the first rotation input corresponding to an indication that the multi-screen device has been rotated a first time from a first open state to a different second open state, wherein data from a first application is displayed while the device is in the first open state on a first screen of the device and not on a second screen of the device, and wherein data from the first application is displayed on the first and second screens of the device in the second open state;

instructions configured to receive a second rotation input, the second rotation input corresponding to an indication that the device has been rotated a second time afrer the first time;

instructions configured to reference display rules for the first application afrer receiving the second rotation input; and

instructions configured to render a display of the first application on one of the first and second screens according to the display rules for the first application.

11. The computer-readable medium of claim 10, wherein the display rules for the first application are stored prior to receiving the second rotation input and correspond to a display configuration of the first application when the device is in the first open state prior to receiving the first rotation input.

12. The method of claim 11, wherein the display rules for the first application indicate that the data from the first application is presented substantially identically to the way in which the data from the first application is presented prior to receiving the first rotation input.

13. The computer-readable medium of claim 10, wherein at least one of the first and second rotation inputs are received from an accelerometer.

14. The computer-readable medium of claim 10, further comprising:

instructions configured to receive a first stack-altering input afrer the first rotation input but before the second rotation input; and

instructions configured to update the display rules for the first application to correspond to display rules that are different from display rules for the first application prior to receiving the first rotation input.

15. The computer-readable medium of claim 14, wherein the first stack-altering input corresponds to at least one of a minimization instruction for the first application and an application-navigation instruction.

16. The computer-readable medium of claim 10, wherein data from a first desktop is displayed on the second screen prior to receiving the first rotation input, wherein the data from the first desktop is not displayed after receiving the first rotation input but before receiving the second rotation input, wherein the data from the first desktop is re-displayed on the second screen after receiving the second rotation input, and wherein the data from the first application is automatically maximized and displayed across both the first and second screens afrer receiving the first rotation input.

17. A multi-screen user device, comprising:

a first screen including a first display area;

a second screen including a second display area;

a sensor configured to produce an electrical signal indicative of a rotation of the user device; and

a computer-readable medium having instructions stored thereon that include:

a first set of instructions configured to receive a first rotation input, the first rotation input corresponding to an indication that the multi-screen device has been rotated a first time from a first open state to a different second open state, wherein data from a first application is displayed while the device is in the first open state on a first screen of the device and not on a second screen of the device, and wherein data from the first application is displayed on the first and second screens of the device in the second open state;

a second set of instructions configured to receive a second rotation input, the second rotation input corresponding to an indication that the device has been rotated a second time afrer the first time;

a third set of instructions configured to reference display rules for the first application afrer receiving the second rotation input; and

a fourth instructions configured to render a display of the first application on one of the first and second screens according to the display rules for the first application.

18. The device of claim 17, wherein the display rules for the first application are stored prior to receiving the second rotation input and correspond to a display configuration of the first application when the device is in the first open state prior to receiving the first rotation input and wherein the display rules for the first application indicate that the data from the first application is presented substantially identically to the way in which the data from the first application is presented prior to receiving the first rotation input.

19. The device of claim 17, wherein the first open state corresponds to an open portrait state and wherein the second state corresponds to an open landscape state.

20. The device of claim 19, wherein the data from the first application is automatically maximized across both the first and second screens in response to receiving the first rotation input.

ABSTRACT

A multi-screen user device and methods for controlling data displayed thereby are disclosed. Specifically, a gesture sequence is disclosed which enables a user to toggle or shift though applications that are displayed by the multi-screen user device. The gesture sequence may correspond to various rotation or partial rotations of the multi-screen user device.

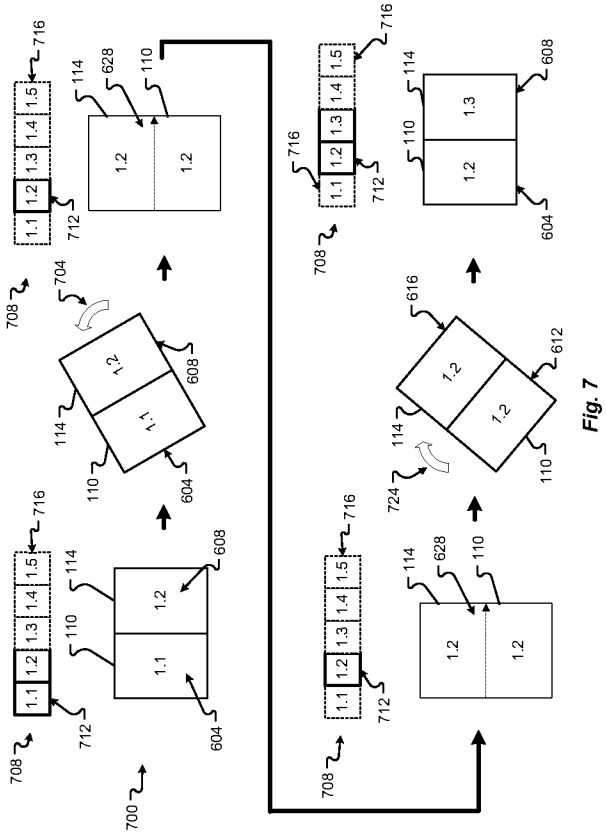

716
1.5
1.4
1.3
1.2
1.1
712
114
628
110
1.2
1.2
608
604
704
708
700
616
612
724

Fig. 7

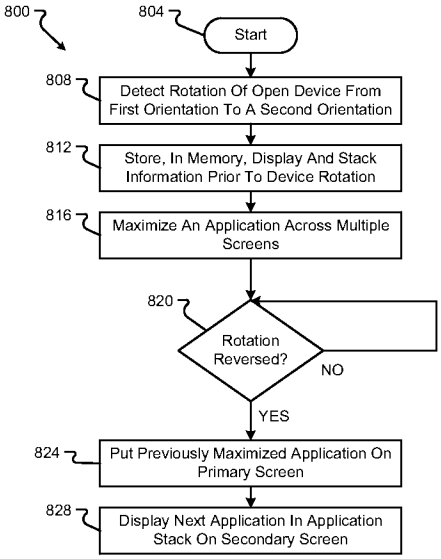

800
804
Start
808
Detect Rotation Of Open Device From First Orientation To A Second Orientation
812
Store, In Memory, Display And Stack Information Prior To Device Rotation
816
Maximize An Application Across Multiple Screens
820
Rotation Reversed?
NO
YES
824
Put Previously Maximized Application On Primary Screen
828
Display Next Application In Application Stack On Secondary Screen

Fig. 8

MODALS IN DUAL DISPLAY COMMUNICATION DEVICES

[0100] The device 100 manages desktops and/or windows with at least one window stack 1700, 1728, as shown in Figs. 9A and 9B. A window stack 1700, 1728 is a logical arrangement of active and/or inactive windows for a multi-screen device. For example, the window stack 1700, 1728 may be logically similar to a deck of cards, where one or more windows or desktops are arranged in order, as shown in Figs. 9A and 9B. An active window is a window that is currently being displayed on at least one of the touch sensitive displays 110, 114. For example, windows 104 and 108 are active windows and are displayed on touch sensitive displays 110 and 114. An inactive window is a window that was opened and displayed but is now "behind" an active window and not being displayed. In embodiments, an inactive window may be for an application that is suspended, and thus, the window is not displaying active content. For example, windows 1712, 1716, 1720, and 1724 are inactive windows.

[0101] A window stack 1700, 1728 may have various arrangements or organizational structures. In the embodiment shown in Fig. 9A, the device 100 includes a first stack 1760 associated with a first touch sensitive display 110 and a second stack associated with a second touch sensitive display 114. Thus, each touch sensitive display 110, 114 can have an associated window stack 1760, 1764. These two window stacks 1760, 1764 may have different numbers of windows arranged in the respective stacks 1760, 1764. Further, the two window stacks 1760, 1764 can also be identified differently and managed separately. Thus, the first window stack 1760 can be arranged in order from a first window 1704 to a next window 1720 to a last window 1724 and finally to a desktop 1722, which, in embodiments, is at the "bottom" of the window stack 1760. In embodiments, the desktop 1722 is not always at the "bottom" as application windows can be arranged in the window stack below the desktop 1722, and the desktop 1722 can be brought to the "top" of a stack over other windows during a desktop reveal. Likewise, the second stack 1764 can be arranged from a first window 1708 to a next window 1712 to a last window 1716, and finally to a desktop 1718, which, in embodiments, is a single desktop area, with desktop 1722, under all the windows in both window stack 1760 and window stack 1764. A logical data structure for managing the two window stacks 1760, 1764 may be as described in conjunction with Fig. 10.

[0102] Another arrangement for a window stack 1728 is shown in Fig. 9B. In this embodiment, there is a single window stack 1728 for both touch sensitive displays 110, 114.

a function of the orientation of the device 100, the context of what programs, functions, software, etc. are being executed on the device 100, how the stack is positioned when the new window is opened, etc. To insert the window in the stack, the position in the stack for the window is determined and the touch sensitive display 110, 114 to which the window is associated may also be determined. With this information, a logical data structure for the window can be created and stored. When user interface or other events or tasks change the arrangement of windows, the window stack(s) can be changed to reflect the change in arrangement. It should be noted that these same concepts described above can be used to manage the one or more desktops for the device 100.

[0106] A logical data structure 1800 for managing the arrangement of windows or desktops in a window stack is shown in Fig. 10. The logical data structure 1800 can be any data structure used to store data whether an object, record, file, etc. The logical data structure 1800 can be stored in any type of database or data storage system, regardless of protocol or standard. In embodiments, the logical data structure 1800 includes one or more portions, fields, attributes, etc. that store data in a logical arrangement that allows for easy storage and retrieval of the information. Hereinafrer, these one or more portions, fields, attributes, etc. shall be described simply as fields. The fields can store data for a window identifier 1804, dimensions 1808, a stack position identifier 1812, a display identifier 1816, and/or an active indicator 1820. Each window in a window stack can have an associated logical data structure 1800. While only a single logical data structure 1800 is shown in Fig. 10, there may be more or fewer logical data structures 1800 used with a window stack (based on the number of windows or desktops in the stack), as represented by ellipses 1824. Further, there may be more or fewer fields than those shown in Fig. 10, as represented by ellipses 1828.

[0107] A window identifier 1804 can include any identifier (ID) that uniquely identifies the associated window in relation to other windows in the window stack. The window identifier 1804 can be a globally unique identifier (GUID), a numeric ID, an alphanumeric ID, or other type of identifier. In embodiments, the window identifier 1804 can be one, two, or any number of digits based on the number of windows that can be opened. In alternative embodiments, the size of the window identifier 1804 may change based on the number of windows opened. While the window is open, the window identifier 1804 may be static and remain unchanged.

Thus, the window stack 1728 is arranged from a desktop 1758 to a first window 1744 to a last window 1756. A window can be arranged in a position among all windows without an association to a specific touch sensitive display 110, 114. In this embodiment, a window is in the order of windows. Further, at least one window is identified as being active. For example, a single window may be rendered in two portions 1732 and 1736 that are displayed on the first touch sensitive screen 110 and the second touch sensitive screen 114. The single window may only occupy a single position in the window stack 1728 although it is displayed on both displays 110, 114.

[0103] Yet another arrangement of a window stack 1760 is shown in Figs. 9C through 9E. The window stack 1760 is shown in three "elevation" views. In Fig. 9C, the top of the window stack 1760 is shown. Two sides of the window stack 1760 are shown in Figs. 9D and 9E. In this embodiment, the window stack 1760 resembles a stack of bricks. The windows are stacked on each other. Looking from the top of the window stack 1760 in Fig, 9C, only the top most windows in the window stack 1760 are seen in different portions of the composite display 1764. The composite display 1764 represents a logical model for the entire display area of the device 100, which can include touch sensitive display 110 and touch sensitive display 114. A desktop 1786 or a window can occupy part or all of the composite display 1764.

[0104] In the embodiment shown, the desktop 1786 is the lowest display or "brick" in the window stack 1760. Thereupon, window 1 1782, window 2 1782, window 3 1768, and window 4 1770 are layered. Window 1 1782, window 3 1768, window 2 1782, and window 4 1770 only occupy a portion of the composite display 1764. Thus, another part of the stack 1760 includes window 8 1774 and windows 5 through 7 shown in section 1790. Only the top window in any portion of the composite display 1764 is actually rendered and displayed. Thus, as shown in the top view in Fig. 9C, window 4 1770, window 8 1774, and window 3 1768 are displayed as being at the top of the display in different portions of the window stack 1760. A window can be dimensioned to occupy only a portion of the composite display 1760 to "reveal" windows lower in the window stack 1760. For example, window 3 1768 is lower in the stack than both window 4 1770 and window 8 1774 but is still displayed. A logical data structure to manage the window stack can be as described in conjunction with Fig. 10.

[0105] When a new window is opened, the newly activated window is generally positioned at the top of the stack. However, where and how the window is positioned within the stack can be

[0108] Dimensions 1808 can include dimensions for a window in the composite display 1760. For example, the dimensions 1808 can include coordinates for two or more corners of the window or may include one coordinate and dimensions for the width and height of the window. These dimensions 1808 can delineate what portion of the composite display 1760 the window may occupy, which may the entire composite display 1760 or only part of composite display 1760. For example, window 4 1770 may have dimensions 1880 that indicate that the window 1770 will occupy only part of the display area for composite display 1760, as shown in Figs. 9C through 9E. As windows are moved or inserted in the window stack, the dimensions 1808 may change.

[0109] A stack position identifier 1812 can be any identifier that can identify the position in the stack for the window or may be inferred from the window's control record within a data structure, such as a list or a stack. The stack position identifier 1812 can be a GUID, a numeric ID, an alphanumeric ID, or other type of identifier. Each window or desktop can include a stack position identifier 1812. For example, as shown in Fig. 9A, window 1 1704 in stack 1 1760 can have a stack position identifier 1812 of 1 identifying that window 1704 is the first window in the stack 1760 and the active window. Similarly, window 6 1724 can have a stack position identifier 1812 of 3 representing that window 1724 is the third window in the stack 1760. Window 2 1708 can also have a stack position identifier 1812 of 1 representing that window 1708 is the first window in the second stack 1764. As shown in Fig. 9B, window 1 1744 can have a stack position identifier 1812 of 1, window 3, rendered in portions 1732 and 1736, can have a stack position identifier 1812 of 3, and window 6 1756 can have a stack position identifier 1812 of 6. Thus, depending on the type of stack, the stack position identifier 1812 can represent a window's location in the stack.

[0110] A display identifier 1816 can identify that the window or desktop is associated with a particular display, such as the first display 110 or the second display 114, or the composite display 1760 composed of both displays. While this display identifier 1816 may not be needed for a multi-stack system, as shown in Fig. 9A, the display identifier 1816 can indicate whether a window in the serial stack of Fig. 9B is displayed on a particular display. Thus, window 3 may have two portions 1732 and 1736 in Fig. 9B. The first portion 1732 may have a display identifier 1816 for the first display while the second portion 1736 may have a display identifier 1816 for the second display 114. However, in alternative embodiments, the window may have

two display identifier 1816 that represent that the window is displayed on both of the displays 110, 114, or a display identifier 1816 identifying the composite display. In another alternate embodiment, the window may have a single display identifier 1816 to represent that the window is displayed on both of the displays 110, 114.

[0111] Similar to the display identifier 1816, an active indicator 1820 may not be needed with the dual stack system of Fig. 9A, as the window in stack position 1 is active and displayed. In the system of Fig. 9B, the active indicator 1820 can indicate which window(s) in the stack is being displayed. Thus, window 3 may have two portions 1732 and 1736 in Fig. 9. The first portion 1732 may have an active indicator 1820 while the second portion 1736 may also have an active indicator 1820. However, in alternative embodiments, window 3 may have a single active indicator 1820. The active indicator 1820 can be a simple flag or bit that represents that the window is active or displayed.

[0112] An embodiment of a method 900 for creating a window stack is shown in Fig. 11. While a general order for the steps of the method 1900 is shown in Fig. 11. Generally, the method 1900 starts with a start operation 1904 and ends with an end operation 1928. The method 1900 can include more or fewer steps or can arrange the order of the steps differently than those shown in Fig. 11. The method 1900 can be executed as a set of computer-executable instructions executed by a computer system and encoded or stored on a computer readable medium. Hereinafrer, the method 1900 shall be explained with reference to the systems, components, modules, software, data structures, user interfaces, etc. described in conjunction with Figs. 1-10.

[0113] A multi-screen device 100 can receive activation of a window, in step 1908. In embodiments, the multi-screen device 100 can receive activation of a window by receiving an input from the touch sensitive display 110 or 114, the configurable area 112 or 116, a gesture capture region 120 or 124, or some other hardware sensor operable to receive user interface inputs. The processor may execute the Task Management Module 540 may receive the input. The Task Management Module 540 can interpret the input as requesting an application task to be executed that will open a window in the window stack.

[0114] In embodiments, the Task Management Module 540 places the user interface interaction in the task stack 552 to be acted upon by the Display Configuration Module 568 of the Multi-Display Management Module 524. Further, the Task Management Module 540 waits

for information from the Multi-Display Management Module 524 to send instructions to the Window Management Module 532 to create the window in the window stack.

[0115] The Multi-Display Management Module 524, upon receiving instruction from the Task Management Module 540, determines to which touch portion of the composite display 1760, the newly activated window should be associated, in step 1912. For example, window 4 1770 is associated with the a portion of the composite display 1764 In embodiments, the device state module 574 of the Multi-Display Management Module 524 may determine how the device is oriented or in what state the device is in, e.g., open, closed, portrait, etc. Further, the preferences module 572 and/or requirements module 580 may determine how the window is to be displayed. The gesture module 576 may determine the user's intentions about how the window is to be opened based on the type of gesture and the location of where the gesture is made.

[0116] The Display Configuration Module 568 may use the input from these modules and evaluate the current window stack 1760 to determine the best place and the best dimensions, based on a visibility algorithm, to open the window. Thus, the Display Configuration Module 568 determines the best place to put the window at the top of the window stack 1760, in step 1916. The visibility algorithm, in embodiments, determines for all portions of the composite display, which windows are at the top of the stack. For example, the visibility algorithm determines that window 3 1768, window 4 1770, and window 8 1774 are at the top of the stack 1760 as viewed in Figs. 9C through 9E. Upon determining where to open the window, the Display Configuration Module 568 can assign a display identifier 816 and possibly dimensions 808 to the window. The display identifier 816 and dimensions 808 can then be sent back to the Task Management Module 540. The Task Management Module 540 may then assign the window a stack position identifier 812 indicating the windows position at the top of the window stack.

[0117] In embodiments, the Task Management Module 540 sends the window stack information and instructions to render the window to the Window Management Module 532. The Window Management Module 532 and the Task Management Module 540 can create the logical data structure 800, in step 1924. Both the Task Management Module 540 and the Window Management Module 532 may create and manage copies of the window stack. These copies of the window stack can be synchronized or kept similar through communications between the Window Management Module 532 and the Task Management Module 540. Thus,

the Window Management Module 532 and the Task Management Module 540, based on the information determined by the Multi-Display Management Module 524, can assign dimensions 808, a stack position identifier 812 (e.g., window 1 1782, window 4 1770, etc.), a display identifier 816 (e.g., touch sensitive display 1 110, touch sensitive display 2 114, composite display identifier, etc,), and an active indicator 820, which is generally always set when the window is at the "top" of the stack. The logical data structure 800 may then be stored by both the Window Management Module 532 and the Task Management Module 540. Further, the Window Management Module 532 and the Task Management Module 540may thereinafrer manage the window stack and the logical data structure(s) 800.

[0118] FIG. 12 depicts a further window stacking configuration. A plurality of windows 1, 2, 3, 4, 5, 6, 7, and 8, whether from the same or different multi-screen or single-screen applications, are depicted. The touch sensitive display 110 currently has window 4 in the active display position while touch sensitive display 114 currently has window 5 in the active display position. The stack for touch sensitive display 110, from top to bottom, has window 4 in the active display position and windows 3, 2, and 1 positioned behind it. The stack for touch sensitive display 114, from top to bottom, has window 5 in the active display position and windows 6, 7, and 8 positioned behind it.

[0119] The desktops D1, D2, D3, D4, D5, and D6 are positioned behind the window stacks. The desktops can be viewed as a desktop stack different from the window stack. Viewed in this manner, the touch sensitive display 110 has a corresponding desktop stack comprising desktops D3, D2, and D1,with desktop D1 being at the bottom 2300 stack position and with desktop D3 at the top stack position being capable of being displayed with window 4 (depending on the window position and size (whether maximized or minimized), and the touch sensitive display 114 has a corresponding desktop stack has a corresponding desktop stack comprising desktops D4, D5, and D6,with desktop D6 being at the bottom 2304 stack position and with desktop D4 at the top stack position being capable of being displayed with window 5 (depending on the window position and size (whether maximized or minimized). Conceptually, the desktops can be viewed as a canvas divided, in this example, into six sections of which two can be displayed, at any one time, the touch sensitive displays 110, 114. When the device 100 is in the closed state, this conceptual model, in one configuration, persists. In this configuration, only one window and desktop stacks can be seen (that corresponding to the primary screen) but the other

window and desktop stacks are virtual; that is, they are maintained in memory but cannot be seen because the secondary screen is not enabled.

[0120] Single-screen applications are typically related only to one window while multi-screen applications can be related to one or more windows. The latter hierarchical relationship is depicted in Fig. 8. As shown in Fig. 8, first window 1 is a root window, second and third windows 1.1 and 1.1.1 are node windows, and fourth and fifth windows 1.1.1.1 and 1.1.2 are leaf windows. The windows are in a hierarchical tree-like sequence and are controlled by a common multi-screen application. The root first window 1 may be a top level view of the hierarchical application window sequence such that there is no parent window corresponding to the root window. The root first window 1 may be a parent to a node window, such as second window 1.1 and the second window 1.1 is a child of the first window 1. One or more node windows may be provided that are related as parent/children. A node window may also serve as a leaf window. By leaf window, it is meant that the leaf window has no corresponding node screen for which the leaf window is a parent (or a leaf window has no child windows). As such, the leaf window does not have any children node windows, Fourth and fifth windows 1.1.1.1 and 1.1.2 are examples of leaf windows. The root view refers to all of the windows controlled by a multi-screen application at a point in time.

[0121] A number of examples will now be discussed with reference to minimizing and maximizing root view windows and subcomponents thereof. Prior to discussing the examples, some fundamental rules regarding maximizing and minimizing displayed images will be discussed. A maximize or minimize operation is typically initiated by a user gesture, particularly a flip or drag. In a drag gesture, a 1:1 map reaction between the user's digit and the displayed image is used as the user "drags" the displayed image to the other touch sensitive display. The user typically releases the displayed image before it is fully moved to the other touch sensitive display. The initial speed after release is generally decreased progressively or gradually until the displayed image is fully located in the desired position. Too little displacement of the displayed image generally will not cause the image to be relocated. Such a low level of displacement is assumed to be accidental or incidental to the user using the device.

[0122] In a root view, or set of hierarchically related windows, the user, when the device is in dual screen mode, can maximize the related windows of the multi-screen application to enable side-by-side viewing of the related windows (e.g., the parent and child windows), thereby

viewing the window hierarchy as a linear progression, with the top-most window being displayed on the left and the lowest-most or deepest level window being displayed on the right (or vice versa). As discussed below, this visual configuration can be controlled using directional transitions conveying a navigation direction and location of a window in relation to a launching point of the window.

[0123] Multi-screen applications can contain modal views on one or multiple windows concurrently. Modals can be initiated automatically or manually. Automatic modals are generally initiated by individual applications. System modals are classified as special modal views brought up by the system, not individual applications. Generally speaking, system modals are used for extremely important, system-level information that requires user confirmation. Examples of system modals include the power-menu and battery warnings. When a system modal is brought up, applications in view are dimmed and disabled, and the system modal is shown above them; this also applies to the desktop. Gestures are disabled when system modals are shown. Of the hardware buttons, both back and home buttons dismiss the modal. This also applies to long-pressing the home button. Closing the device will result in the system modal being moved to the screen that remains active. Incoming calls cancel system modals.

[0124] As a general rule, modals are always launched on the view they were initiated from. Modals dim out the screen they obscure, as well as the other view if launched on top of a maximized application. When full-screen modals are provided by other applications (such as Email providing its compose view to any other application that wants to compose an email), the result of modal launching depends on the capabilities of both the application that launched the modal and the application that provides the modal.

[0125] Note that the exact positioning of system modals in the wireframes does not necessarily represent the actual positioning on the screens. The visual design document provides details on system modal positioning.

[0126] A first example will be applied to Figs. 13A-C. This example shows that pop-up modals remain in view whenever a window of an application is maximized. With reference to Fig. 13A, the first and second touch sensitive displays 110, 114 display, respectively, a (minimized) node second window 1.1 (1304) and a second desktop D2. A pop-up modal 1350 is displayed on top of the second window 1.1 (1304). Stated another way, the first touch sensitive display 110 has the second window 1.1 (1304) in an active display position, and the second touch

sensitive display 114 has the second desktop D2 in an active display position. In a gesture capture region 120 or 124 (or in other configurations the first or second touch sensitive display 110 or 114) receives, from the user, a gesture 1300. Exemplary gestures include a drag or flick to the right (though one or more other gestures may be employed). By the gesture 1300, the user seeks to maximize the root view to show the second window 1.1 (1304) and a node third window 1.1.1 (1308) in separate displays (Fig. 13C). Thus, the third window 1.1.1 (1308) moves to the active display position on the second touch sensitive display 114, and the node second window 1.1 (1304) remains in the active display position on the first touch sensitive display 110 as shown by Fig. 13C. The second desktop D2, meanwhile, moves to an inactive display position on the second touch sensitive display 114. The third window 1.1.1 (1308) does this by sliding over to the second touch sensitive display 114 from under the second window 1.1 (1304). As can be seen from Figs. 13A-C, the pop-up modal 1350 remains in the window in the hierarchy from which it was initiated (namely the second window 1.1).

[0127] A second example will be discussed with reference to Figs. 14A-C in which the first touch sensitive display 110 has a first desktop D1 in an active display position, and the second touch sensitive display 114 has the leaf fifth window 1.1.1.1 (1312) in an active display position. A pop-up modal 1350 has been launched in the fifth window 1.1.1.1 (1312). In a gesture capture region 120 or 124 (or in other configurations the first or second touch sensitive display 110 or 114) receives, from the user, a gesture 1300, such as a flick or drag, to the left. By the gesture 1300, the user seeks to maximize the root view to show the node third window 1.1.1 (1308) and the leaf fifth window 1.1.1.1 (1312) in separate displays (Fig. 14B). As a general rule, a leaf window (or window having no child) is positioned on the second touch sensitive display 114. Thus, the leaf fifth window 1.1.1.1 (1312) remains in the active display position on the second touch sensitive display 114, and the node third window 1.1.1 (1308) is moved to the active display position on the first touch sensitive display 110 as shown by Fig. 14C. The first desktop D1, meanwhile, moves to an inactive display position on the first touch sensitive display 110. The node third window 1.1.1 (1308) does this by sliding from under the leaf fifth window 1.1.1.1 (1312). As in the prior example, the pop-up modal 1350 remains in the window in the hierarchy from which it was initiated (namely the fifth window 1.1.1.1).

[0128] A third example will be discussed with reference to Figs. 15A-C in which the first touch sensitive display 110 displays the leaf fifth window 1.1.1.1 (1312), and the second touch

sensitive display 114 displays the second desktop D2. A modal 1350 has been launched in the fifth window 1.1.1.1 (1312). In a gesture capture region 120 or 124 (or in other configurations the first or second touch sensitive display 110 or 114) receives, from the user, a gesture 1300, particularly a flick gesture, to the right. By the gesture 1300, the user seeks to move the leaf fifth window 1.1.1.1 (1312) to the second touch sensitive display 114 to reveal the first desktop D1. As a general rule, a flick gesture will move the leaf fifth window 1.1.1.1 (1312) to the other touch sensitive display. Thus, the leaf fifth window 1.1.1.1 (1312) moves to the active display position on the second touch sensitive display 114, and the first desktop D1 moves to the active display position on the first touch sensitive display 110 as shown by Fig. 15C. The second desktop D2, meanwhile, moves to an inactive display position on the second touch sensitive display 110. As in the prior example, the modal 1350 remains in the window in the hierarchy from which it was initiated (namely the fifth window 1.1.1.1).

[0129] A fourth example will be discussed with reference to Figs. 16A-C in which the first touch sensitive display 110 displays a first desktop D1, and the second touch sensitive display 114 displays a node second window 1.1 (1304) (with a node third window 1.1.1 (1308) and fifth window 1.1.1.1 (1312) in inactive display positions). A modal 1350 has been launched in the second touch sensitive display 114. In a gesture capture region 120 or 124 (or in other configurations the first or second touch sensitive display 110 or 114) receives, from the user, a gesture 1300, such as a flick or drag, to the left. By the gesture 1300, the user seeks to maximize the root view to show the second window 1.1 (1304) and third window 1.1.1 (1308) in separate displays (Fig. 16C). As a general rule, whenever a child window (e.g., a node or leaf window) is uncovered through maximization, it is uncovered on the second touch sensitive display 114. Thus, the second window 1.1 (1304) moves to an active display position in the first touch sensitive display 110 while the third window 1.1.1 (1308) moves to an active display position on the second touch sensitive display 114 as shown by Fig. 16C. The first desktop D1, meanwhile, moves to an inactive display position on the first touch sensitive display 110. Due to technical restrictions (e.g., when pop-up modals are launched as tasks), the pop-up modal 1350 is canceled during the maximization operation.

[0130] Generally, pop-up modals 1350 remain on the display they were initiated from unless the window that contains the modal 1350 is hidden during minimization. A fourth example will be discussed with reference to Figs. 17A-C in which the first touch sensitive display 110 has the

second window 1.1 (1304) in an active display position, and the second touch sensitive display 114 has the third window 1.1.1 (1308) in an active display position. A pop-up modal 1350 has been launched in the second window 1.1 (1304). In a gesture capture region 120 or 124 (or in other configurations the first or second touch sensitive display 110 or 114) receives, from the user, a gesture 1300, such as a flick or drag, to the left. By the gesture 1300, the user seeks to minimize the root view to show the second window 1.1 (1304) in the first touch sensitive display 110, and the second desktop D2 in the second touch sensitive display 114 (Fig. 17C). Thus, the second window 1.1 (1304) remains in the active display position on the second touch sensitive display 110, while the third window 1.1.1 (1308) slides under the second window 1.1 (1304) as shown in Fig. 17B. The modal 1350 remains in view on the second window 1.1 (1304) as the second window has not been hidden during minimization. The second desktop D2, meanwhile, moves to an active display position on the second touch sensitive display 114.

[0131] A fifth example will be discussed with reference to Figs. 18A-C in which the first touch sensitive display 110 displays the node third window 1.1.1 (1308), and the second touch sensitive display 114 displays the leaf fifth window 1.1.1.1 (1312). A modal 1350 has been launched on the fifth window 1.1.1.1 (1312). In a gesture capture region 120 or 124 (or in other configurations the first or second touch sensitive display 110 or 114) receives, from the user, a gesture 1300, particularly a drag or flick gesture, to the right. By the gesture 1300, the user seeks to move the third window 1.1.1 (1308) to the second touch sensitive display 114 to reveal the first desktop D1. Thus, the third window 1.1.1 (1308) moves to the active display position on the second touch sensitive display 114, and the first desktop D1 moves to the active display position on the first touch sensitive display 110 as shown by Fig. 18C. The fifth window 1.1.1.1 (1312), meanwhile, moves to an inactive display position on the second touch sensitive display 110. Because the modal 1350 was launched from the fifth window 1.1.1.1 (1312), the modal is hidden or canceled with the fifth window 1.1.1.1 (1312) as shown in Fig. 18B.

[0132] A sixth example will be discussed with reference to Figs. 19A-C in which the first touch sensitive display 110 displays the node third window 1.1.1 (1308), and the second touch sensitive display 114 displays the leaf fifth window 1.1.1.1 (1312). A modal 1350 has been launched in the third window 1.1.1 (1308). In a gesture capture region 120 or 124 (or in other configurations the first or second touch sensitive display 110 or 114) receives, from the user, a gesture 1300, particularly a drag or flick gesture, to the left. By the gesture 1300, the user seeks

to move the root view to the first touch sensitive display 110 to reveal the second desktop D2 in the second touch sensitive display 114. Thus, the third window 1.1.1 (1308) remains in the active display position on the first touch sensitive display 110, and the second desktop D2 moves to the active display position on the second touch sensitive display 114 as shown by Fig. 19C. The fifth window 1.1.1.1 (1312) , meanwhile, moves to an inactive display position on the first touch sensitive display 110 by sliding under the third window 1.1.1 (1308). Due to technical restrictions (e.g., when the modal 1350 is launched as a task), the modal is canceled during minimization.

[0133] Generally, modals remain on the display they were launched on during and after rotation of the communication device 100. A seventh example will be discussed to Figs. 20A-C. The first and second touch sensitive displays 110, 114 display, respectively, a node second window 1.1 (1304) and a node third window 1.1.1 (1308) (Fig. 20A). Stated another way, the first touch sensitive display 110 has the second window 1.1 (1304) in an active display position, and the second touch sensitive display 114 has the third window 1.1.1 (1308) in an active display position. A modal 1350 has been launched in the first touch sensitive display 110. The user rotates the communication clockwise (Fig. 20B) to transition the communication device from portrait display mode to landscape display mode (Fig. 20C). As a result of rotation, the second window 1.1 (1304) has expanded to cover both the first and second touch sensitive displays 110 and 114 (Fig. 20C). The modal 1350 remains in the first touch sensitive display 110 both before and afrer rotation or remains on the maximized second window 1.1 (1308).

[0134] An eighth example will be discussed with reference to Figs. 21A-C in which the first touch sensitive display 110 has the node second window 1.1 (1304) in an active display position, and the second touch sensitive display 114 has the node third window 1.1.1 (1308) in an active display position. A modal 1350 has been launched in the third window 1.1.1 (1308). The user rotates the communication clockwise (Fig. 21B) to transition the communication device from portrait display mode to landscape display mode (Fig. 21C). As a result of rotation, the second window 1.1 (1304) has expanded to cover both the first and second touch sensitive displays 110 and 114 (Fig. 21C). Since the first touch sensitive display 110 before and after rotation did not contain the modal 1350 and the second window 1.1 (1304) is expanded to fill both the first and second touch sensitive displays 110 and 114, the modal 1350 is canceled.

[0135] A ninth example will be discussed with reference to Figs. 22A-C in which both the first and second touch sensitive displays 110 and 114 display the node second window 1.1 (1304). A modal 1350 has been launched in the first touch sensitive display 110 (which is the upper display). The user rotates the communication counterclockwise (Fig. 22B) to transition the communication device from landscape display mode to portrait display mode (Fig. 22C). As a result of rotation, the third window 1.1.1 (1308) has slid out from under the second window 1.1 (1304) and moved into the active display position in the second touch sensitive display 114 (Fig. 21C). A modal 1350 on a maximized landscape multi-screen application continues to be displayed on the respective window after rotation to portrait display mode.

[0136] A modal, particularly a pop-up modal 1350, is launched on a selected display (step 2300) containing a window of an application. In the various examples, middleware 520, particularly one or more of the Multi-Display Management (MDM) class 524, a Surface Cache class 528, a Window Management class 532, an Activity Management class 536, and an Application Management class 540, independently or collectively, determine, such as by a gesture or rotation of the communication device 100, that a set of windows is to be maximized or minimized (step 2304). The middleware 520 determines whether the set of windows is to be maximized or minimized (decision diamond 2308). When the set of windows is to be maximized, the middleware 520 determines an orientation of the device 100 and the window and display from which the modal was launched and moves the modal with the respective window or display, as appropriate (step 2312). In portrait display mode, the modal is moved with the respective window from which it was launched. In landscape display mode, the modal is moved with the display from which it was launched. When the set of windows is to be minimized, the middleware 520 determines an orientation of the device 100 and the window from which the modal was launched and moves the modal with the respective window or display, as appropriate (step 2316). The modal is canceled if the window is hidden.

[0137] The exemplary systems and methods of this disclosure have been described in relation to communication devices. However, to avoid unnecessarily obscuring the present disclosure, the preceding description omits a number of known structures and devices. This omission is not to be construed as a limitation of the scopes of the claims. Specific details are set forth to provide an understanding of the present disclosure. It should however be appreciated that the

present disclosure may be practiced in a variety of ways beyond the specific detail set forth herein.

[0138] Furthermore, while the exemplary aspects, embodiments, and/or configurations illustrated herein show the various components of the system collocated, certain components of the system can be located remotely, at distant portions of a distributed network, such as a LAN and/or the Internet, or within a dedicated system. Thus, it should be appreciated, that the components of the system can be combined in to one or more devices, such as a communication device, or collocated on a particular node of a distributed network, such as an analog and/or digital telecommunications network, a packet-switch network, or a circuit-switched network. It will be appreciated from the preceding description, and for reasons of computational efficiency, that the components of the system can be arranged at any location within a distributed network of components without affecting the operation of the system. For example, the various components can be located in a switch such as a PBX and media server, gateway, in one or more communications devices, at one or more users' premises, or some combination thereof. Similarly, one or more functional portions of the system could be distributed between a telecommunications device(s) and an associated computing device.

[0139] Furthermore, it should be appreciated that the various links connecting the elements can be wired or wireless links, or any combination thereof, or any other known or later developed element(s) that is capable of supplying and/or communicating data to and from the connected elements. These wired or wireless links can also be secure links and may be capable of communicating encrypted information. Transmission media used as links, for example, can be any suitable carrier for electrical signals, including coaxial cables, copper wire and fiber optics, and may take the form of acoustic or light waves, such as those generated during radio-wave and infra-red data communications.

[0140] Also, while the flowcharts have been discussed and illustrated in relation to a particular sequence of events, it should be appreciated that changes, additions, and omissions to this sequence can occur without materially affecting the operation of the disclosed embodiments, configuration, and aspects.

[0141] A number of variations and modifications of the disclosure can be used. It would be possible to provide for some features of the disclosure without providing others.

[0142] In other embodiments, other rules may be applied. For example, the preferred orientations of the higher and lower level related windows is different from the orientations described above.

[0143] In yet another embodiment, the systems and methods of this disclosure can be implemented in conjunction with a special purpose computer, a programmed microprocessor or microcontroller and peripheral integrated circuit element(s), an ASIC or other integrated circuit, a digital signal processor, a hard-wired electronic or logic circuit such as discrete element circuit, a programmable logic device or gate array such as PLD, PLA, FPGA, PAL, special purpose computer, any comparable means, or the like. In general, any device(s) or means capable of implementing the methodology illustrated herein can be used to implement the various aspects of this disclosure. Exemplary hardware that can be used for the disclosed embodiments, configurations and aspects includes computers, handheld devices, telephones (e.g., cellular, Internet enabled, digital, analog, hybrids, and others), and other hardware known in the art. Some of these devices include processors (e.g., a single or multiple microprocessors), memory, nonvolatile storage, input devices, and output devices. Furthermore, alternative software implementations including, but not limited to, distributed processing or component/object distributed processing, parallel processing, or virtual machine processing can also be constructed to implement the methods described herein.

[0144] In yet another embodiment, the disclosed methods may be readily implemented in conjunction with software using object or object-oriented software development environments that provide portable source code that can be used on a variety of computer or workstation platforms. Alternatively, the disclosed system may be implemented partially or fully in hardware using standard logic circuits or VLSI design. Whether software or hardware is used to implement the systems in accordance with this disclosure is dependent on the speed and/or efficiency requirements of the system, the particular function, and the particular software or hardware systems or microprocessor or microcomputer systems being utilized.

[0145] In yet another embodiment, the disclosed methods may be partially implemented in software that can be stored on a storage medium, executed on programmed general-purpose computer with the cooperation of a controller and memory, a special purpose computer, a microprocessor, or the like. In these instances, the systems and methods of this disclosure can be implemented as program embedded on personal computer such as an applet, JAVA® or CGI

script, as a resource residing on a server or computer workstation, as a routine embedded in a dedicated measurement system, system component, or the like. The system can also be implemented by physically incorporating the system and/or method into a software and/or hardware system.

[0146] Although the present disclosure describes components and functions implemented in the aspects, embodiments, and/or configurations with reference to particular standards and protocols, the aspects, embodiments, and/or configurations are not limited to such standards and protocols. Other similar standards and protocols not mentioned herein are in existence and are considered to be included in the present disclosure. Moreover, the standards and protocols mentioned herein and other similar standards and protocols not mentioned herein are periodically superseded by faster or more effective equivalents having essentially the same functions. Such replacement standards and protocols having the same functions are considered equivalents included in the present disclosure.

[0147] The present disclosure, in various aspects, embodiments, and/or configurations, includes components, methods, processes, systems and/or apparatus substantially as depicted and described herein, including various aspects, embodiments, configurations embodiments, subcombinations, and/or subsets thereof. Those of skill in the art will understand how to make and use the disclosed aspects, embodiments, and/or configurations afrer understanding the present disclosure. The present disclosure, in various aspects, embodiments, and/or configurations, includes providing devices and processes in the absence of items not depicted and/or described herein or in various aspects, embodiments, and/or configurations hereof, including in the absence of such items as may have been used in previous devices or processes, e.g., for improving performance, achieving ease and\or reducing cost of implementation.

[0148] The foregoing discussion has been presented for purposes of illustration and description. The foregoing is not intended to limit the disclosure to the form or forms disclosed herein. In the foregoing Detailed Description for example, various features of the disclosure are grouped together in one or more aspects, embodiments, and/or configurations for the purpose of streamlining the disclosure. The features of the aspects, embodiments, and/or configurations of the disclosure may be combined in alternate aspects, embodiments, and/or configurations other than those discussed above. This method of disclosure is not to be interpreted as reflecting an intention that the claims require more features than are expressly recited in each claim. Rather,

as the following claims reflect, inventive aspects lie in less than all features of a single foregoing disclosed aspect, embodiment, and/or configuration. Thus, the following claims are hereby incorporated into this Detailed Description, with each claim standing on its own as a separate preferred embodiment of the disclosure.

[0149] Moreover, though the description has included description of one or more aspects, embodiments, and/or configurations and certain variations and modifications, other variations, combinations, and modifications are within the scope of the disclosure, e.g., as may be within the skill and knowledge of those in the art, afrer understanding the present disclosure. It is intended to obtain rights which include alternative aspects, embodiments, and/or configurations to the extent permitted, including alternate, interchangeable and/or equivalent structures, functions, ranges or steps to those claimed, whether or not such alternate, interchangeable and/or equivalent structures, functions, ranges or steps are disclosed herein, and without intending to publicly dedicate any patentable subject matter.

What is claimed is:

1. A method, comprising:

receiving, by processor executable middleware in a multi-display communication device, a command to at least one of minimize and maximize a set of related windows from a common multi-screen application, the set of related windows comprising a higher level window and a lower level window at least one of displayed and to be displayed by different touch sensitive displays and at least one of the higher and lower level windows comprises a modal launched from one of the higher and lower level windows;

applying, by the processor executable middleware, the following rules:

(B1) when the set of related windows is to be maximized, maintaining the modal in view in the one of the higher and lower level windows from which the modal was launched; and

(B2) when the set of related windows is to be minimized, applying at least one of the following sub-rules:

(B2i) when the at least one of the higher and lower level windows comprising the modal is displayed afrer the set of related windows is minimized, maintaining the modal in view; and

(B2ii) when the at least one of the higher and lower level windows comprising the modal is not to be displayed after the set of related windows is minimized, cancelling automatically the modal.

2. The method of claim 1, wherein rule (B1) applies and wherein the processor executable middleware selects a predetermined touch sensitive display for displaying one of the higher and lower level windows and causes the other of the higher and lower level windows to be displayed on the other touch sensitive display.

3. The method of claim 2, wherein, prior to maximization, the modal is displayed on the higher level window and wherein the modal was launched from the lower level window.

4. The method of claim 3, wherein a drag or flick command is received, wherein neither the higher nor lower level window is displayed in the predetermined touch sensitive display when the command is received, and wherein the lower level window appears to move from the other touch sensitive display to the predetermined touch sensitive display, thereby revealing the higher level window in the other touch sensitive display.

5. The method of claim 3, wherein a drag or flick command is received, wherein the lower level window is in the predetermined touch sensitive display when the command is received, wherein the higher level window appears to move from the predetermined touch sensitive display to the other touch sensitive display, whereby the lower level window is displayed by the predetermined touch sensitive display and the higher level window is displayed by the other touch sensitive display.

6. The method of claim 1, wherein rule (B2i) applies.

7. The method of claim 6, wherein rule (B2ii) applies.

8. The method of claim 1, wherein rule (B2) applies, wherein the processor executable middleware selects a touch sensitive display for displaying the higher level window and ceases to display the lower level window on the other touch sensitive display, wherein a drag or flick command is received, wherein the higher and lower level window are concurrently displayed in the selected and other touch sensitive displays when the command is received, wherein the lower level display is displayed in the selected touch sensitive display when the command is received, and wherein the higher level window appears to move from the other touch sensitive display to the selected touch sensitive display, thereby covering the lower level window in the other touch sensitive display.

9. The method of claim 1, wherein rule (B2) applies, wherein the processor executable middleware selects a touch sensitive display for displaying the higher level window and ceases to display the lower level window on the other touch sensitive display, wherein a drag or flick command is received, wherein the higher and lower level window are concurrently displayed in the selected and other touch sensitive displays when the command is received, wherein the higher level display is displayed in the selected touch sensitive display when the command is received, and wherein the lower level window appears to move from the other touch sensitive display to the selected touch sensitive display, thereby disappearing under the higher level window in the other touch sensitive display.

10. A non-transient and tangible computer readable medium comprising processor executable instructions to perform the steps of claim 1.

11. A multi-display communication device operable to perform the steps of claim 1.

12. A method, comprising:

receiving, by processor executable middleware in a multi-display communication device comprising first and second touch sensitive displays, a command to rotate a display from one of portrait and landscape display mode to the other of portrait and landscape display mode, wherein the first touch sensitive display comprises a modal launched from at least one of a higher and lower level window in a set of related windows comprising a higher level window and a lower level window at least one of displayed and to be displayed by different touch sensitive displays;

applying, by the processor executable middleware, the following rules:

(B1) when the first and second touch sensitive displays are rotated from a portrait display mode to a landscape display mode, applying at least one of the following sub-rules:

(B1i) when the first touch sensitive display is, relative to the second touch sensitive display, in an upper position after rotation and the first touch sensitive display displays the at least one of a higher and lower level window, displaying the modal in the first touch sensitive display after rotation; and

(B1ii) when the first touch sensitive display is, relative to the second touch sensitive display, in a lower position after rotation and neither the first nor second touch sensitive display displays the at least one of a higher and lower level window, not displaying, after rotation, the modal in either the first or second touch sensitive display; and

(B2) when the first and second touch sensitive displays are rotated from a landscape display mode to a portrait display mode, displaying the modal, after rotation, on the one of the first and second touch sensitive displays on which the modal was launched.

13. The method of claim 12, wherein the set of related windows is maximized before and after rotation.

14. The method of claim 12, wherein rule (B1i) applies.

15. The method of claim 14, wherein, after rotation, the at least one of a higher and lower level window displaying the modal is displayed on both the first and second touch sensitive displays and wherein, during rotation, a lower level window displayed on the second touch sensitive display is covered by the at least one of a higher and lower level window.

16. The method of claim 14, wherein, before rotation, the at least one of a higher and lower level window displaying the modal is displayed on the second touch sensitive display, wherein, after rotation, the modal is displayed on the first touch sensitive display, and wherein,

after rotation, the at least one of a higher and lower level window is displayed on both the first and second touch sensitive displays.

17. The method of claim 12, wherein rule (B1ii) applies.

18. The method of claim 17, wherein the modal is canceled automatically at least one of during and after rotation.

19. The method of claim 12, wherein rule (B2) applies.

20. A non-transient and tangible computer readable medium comprising processor executable instructions to perform the steps of claim 12.

21. A multi-display communication device operable to perform the steps of claim 12.

ABSTRACT

[0150] The present disclosure is directed to methodologies and devices for handling modals in a set of related windows.

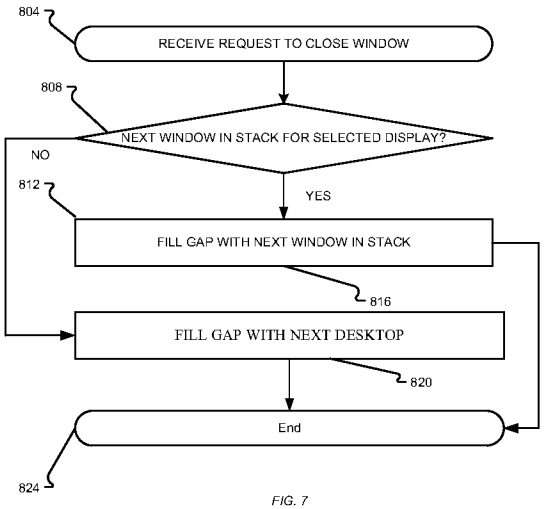


FIG. 7

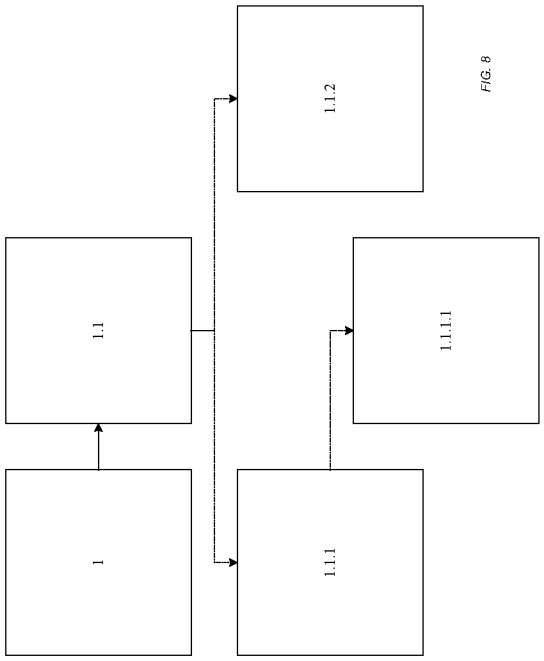


FIG. 8

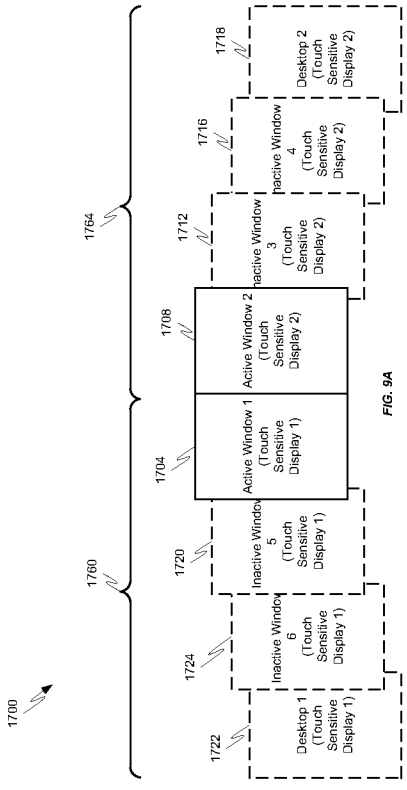


FIG. 9A

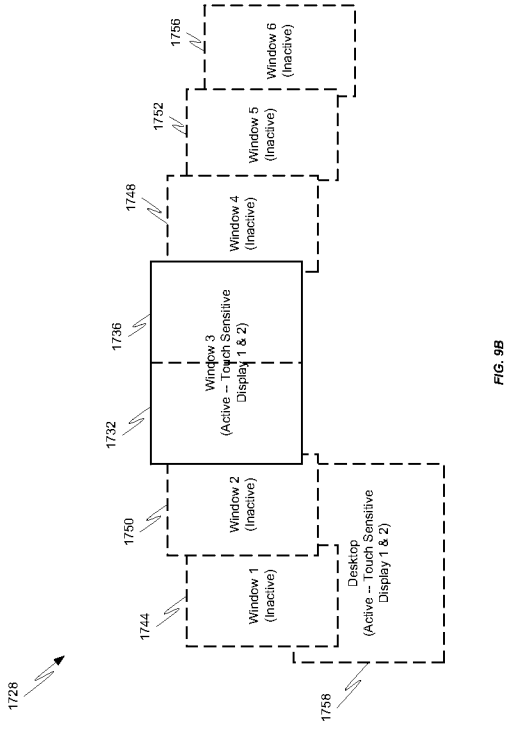


FIG. 9B

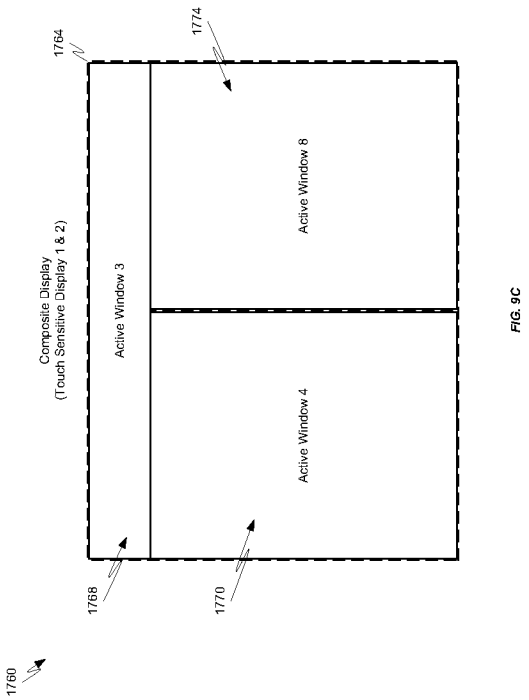


FIG. 9C

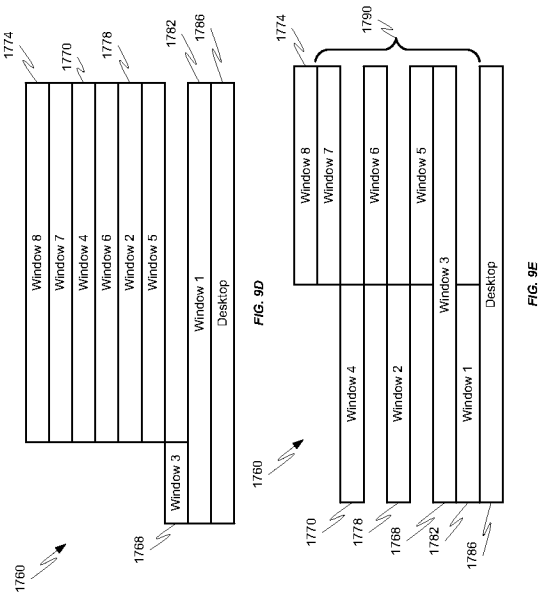


FIG. 9D

FIG. 9E

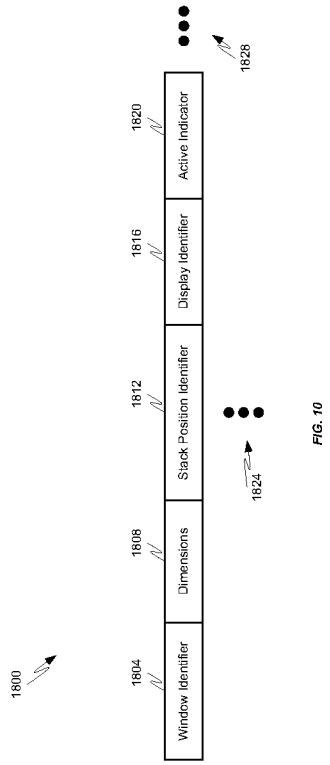


FIG. 10

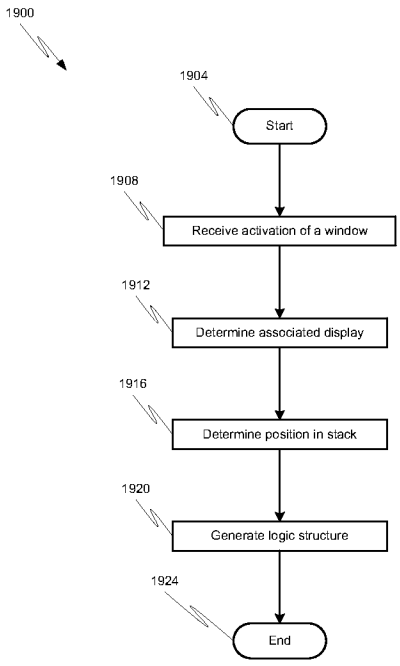


FIG. 11

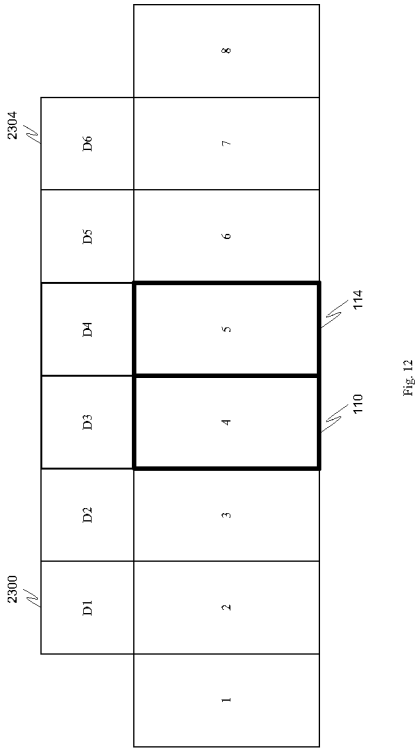


Fig. 12

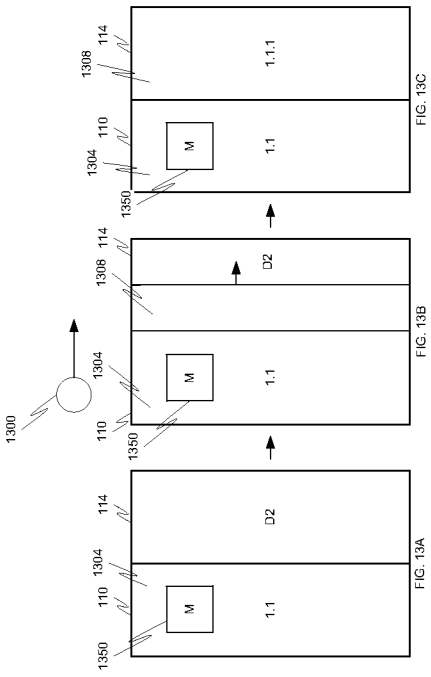


FIG. 13A FIG. 13B FIG. 13C

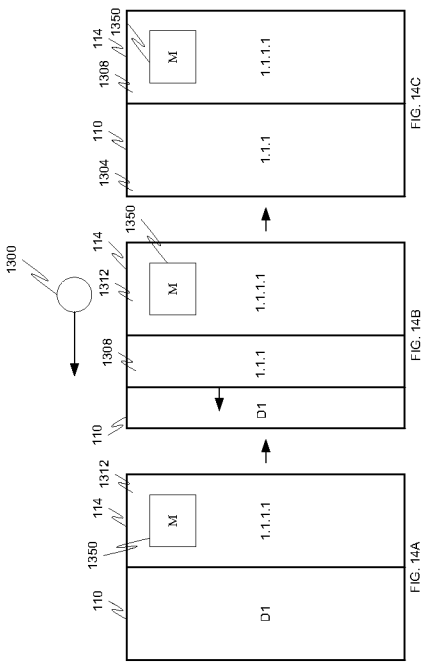


FIG. 14A FIG. 14B FIG. 14C

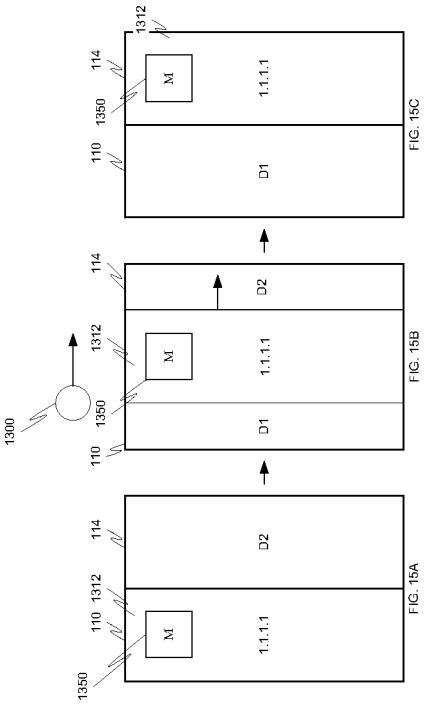
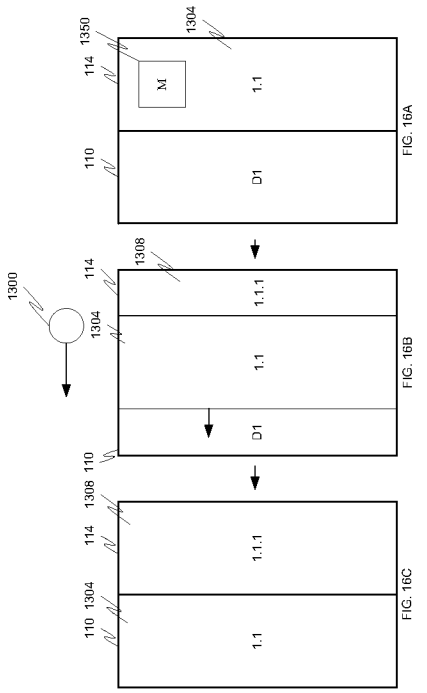
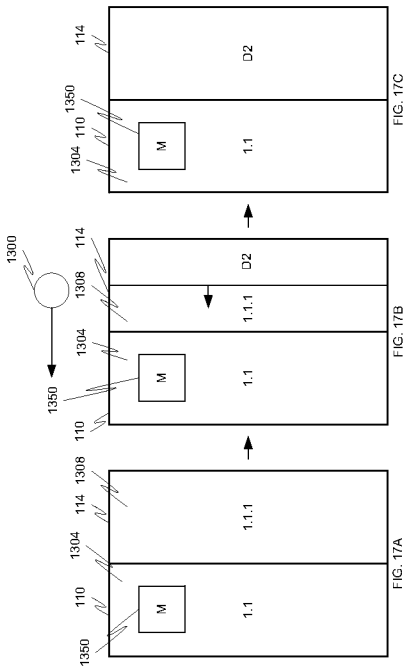
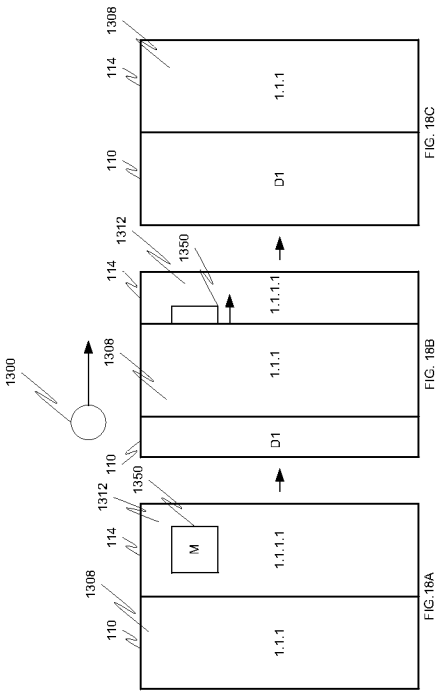
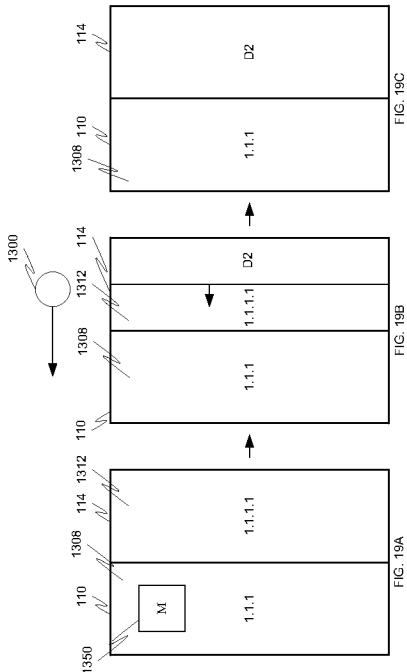


FIG. 15A FIG. 15B FIG. 15C


1350
1304
114
M
1.1
110
D1
1308
1.1.1
1300
1304
1.1
FIG. 16A
FIG. 16B
FIG. 16C


114
D2
1350
1304
110
M
1.1
1.1.1
1308
1300
FIG. 17C
FIG. 17B
FIG. 17A


1308
114
1.1.1
110
D1
1312
1350
1300
M
FIG. 18C
FIG. 18B
FIG.18A


114
D2
1308
110
1.1.1
1312
1300
1350
M
FIG. 19C
FIG. 19B
FIG. 19A

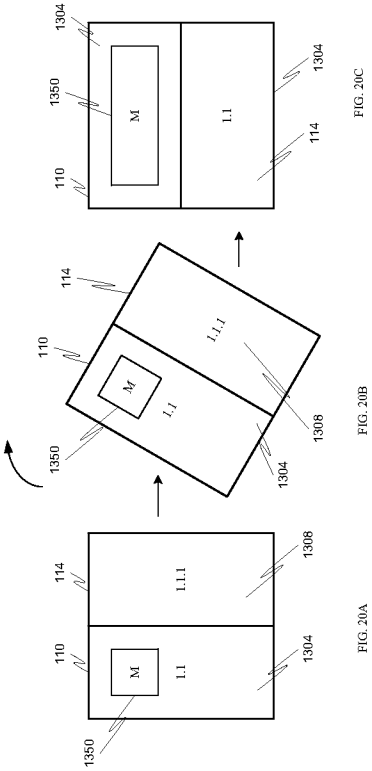


FIG. 20C

FIG. 20B

FIG. 20A

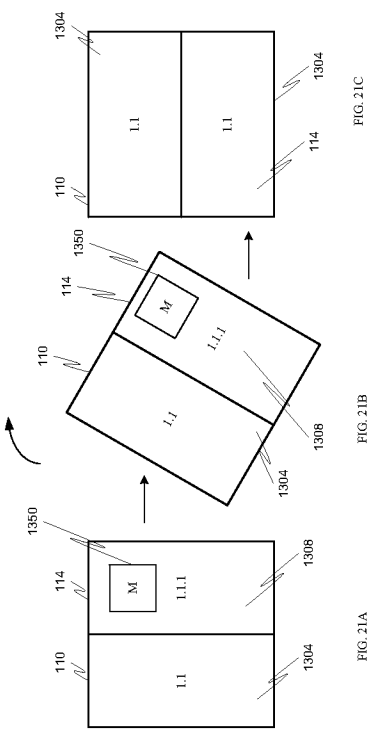


FIG. 21C

FIG. 21B

FIG. 21A

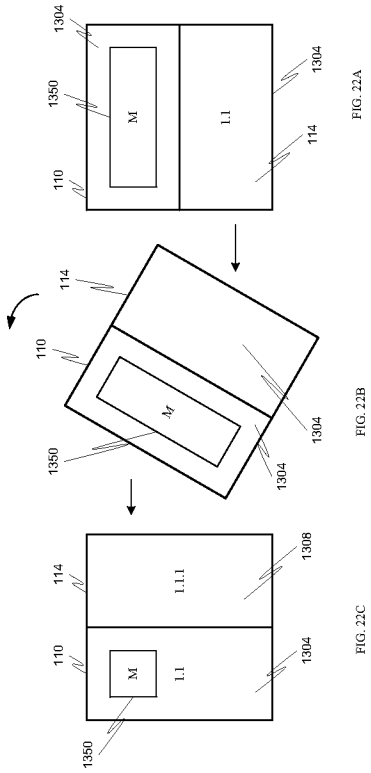


FIG. 22A

FIG. 22B

FIG. 22C

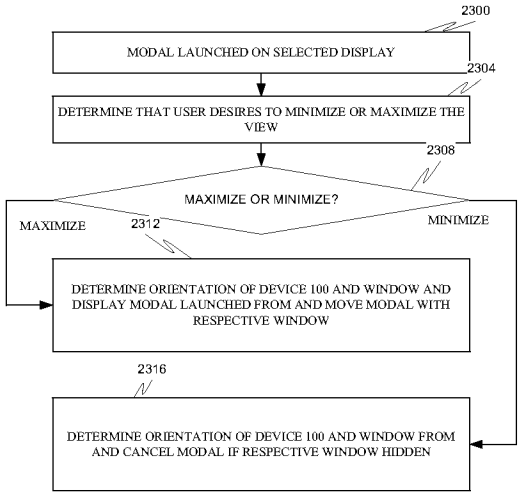


FIG. 23

MULTIPLE CHILD WINDOWS IN DUAL DISPLAY COMMUNICATION DEVICES

[0100] The device 100 manages desktops and/or windows with at least one window stack 1700, 1728, as shown in Figs. 9A and 9B. A window stack 1700, 1728 is a logical arrangement of active and/or inactive windows for a multi-screen device. For example, the window stack 1700, 1728 may be logically similar to a deck of cards, where one or more windows or desktops are arranged in order, as shown in Figs. 9A and 9B. An active window is a window that is currently being displayed on at least one of the touch sensitive displays 110, 114. For example, windows 104 and 108 are active windows and are displayed on touch sensitive displays 110 and 114. An inactive window is a window that was opened and displayed but is now "behind" an active window and not being displayed. In embodiments, an inactive window may be for an application that is suspended, and thus, the window is not displaying active content. For example, windows 1712, 1716, 1720, and 1724 are inactive windows.

[0101] A window stack 1700, 1728 may have various arrangements or organizational structures. In the embodiment shown in Fig. 9A, the device 100 includes a first stack 1760 associated with a first touch sensitive display 110 and a second stack associated with a second touch sensitive display 114. Thus, each touch sensitive display 110, 114 can have an associated window stack 1760, 1764. These two window stacks 1760, 1764 may have different numbers of windows arranged in the respective stacks 1760, 1764. Further, the two window stacks 1760, 1764 can also be identified differently and managed separately. Thus, the first window stack 1760 can be arranged in order from a first window 1704 to a next window 1720 to a last window 1724 and finally to a desktop 1722, which, in embodiments, is at the "bottom" of the window stack 1760. In embodiments, the desktop 1722 is not always at the "bottom" as application windows can be arranged in the window stack below the desktop 1722, and the desktop 1722 can be brought to the "top" of a stack over other windows during a desktop reveal. Likewise, the second stack 1764 can be arranged from a first window 1708 to a next window 1712 to a last window 1716, and finally to a desktop 1718, which, in embodiments, is a single desktop area, with desktop 1722, under all the windows in both window stack 1760 and window stack 1764. A logical data structure for managing the two window stacks 1760, 1764 may be as described in conjunction with Fig. 10.

[0102] Another arrangement for a window stack 1728 is shown in Fig. 9B. In this embodiment, there is a single window stack 1728 for both touch sensitive displays 110, 114.

Thus, the window stack 1728 is arranged from a desktop 1758 to a first window 1744 to a last window 1756. A window can be arranged in a position among all windows without an association to a specific touch sensitive display 110, 114. In this embodiment, a window is in the order of windows. Further, at least one window is identified as being active. For example, a single window may be rendered in two portions 1732 and 1736 that are displayed on the first touch sensitive screen 110 and the second touch sensitive screen 114. The single window may only occupy a single position in the window stack 1728 although it is displayed on both displays 110, 114.

[0103] Yet another arrangement of a window stack 1760 is shown in Figs. 9C through 9E. The window stack 1760 is shown in three "elevation" views. In Fig. 9C, the top of the window stack 1760 is shown. Two sides of the window stack 1760 are shown in Figs. 9D and 9E. In this embodiment, the window stack 1760 resembles a stack of bricks. The windows are stacked on each other. Looking from the top of the window stack 1760 in Fig, 9C, only the top most windows in the window stack 1760 are seen in different portions of the composite display 1764. The composite display 1764 represents a logical model for the entire display area of the device 100, which can include touch sensitive display 110 and touch sensitive display 114. A desktop 1786 or a window can occupy part or all of the composite display 1764.

[0104] In the embodiment shown, the desktop 1786 is the lowest display or "brick" in the window stack 1760. Thereupon, window 1 1782, window 2 1782, window 3 1768, and window 4 1770 are layered. Window 1 1782, window 3 1768, window 2 1782, and window 4 1770 only occupy a portion of the composite display 1764. Thus, another part of the stack 1760 includes window 8 1774 and windows 5 through 7 shown in section 1790. Only the top window in any portion of the composite display 1764 is actually rendered and displayed. Thus, as shown in the top view in Fig. 9C, window 4 1770, window 8 1774, and window 3 1768 are displayed as being at the top of the display in different portions of the window stack 1760. A window can be dimensioned to occupy only a portion of the composite display 1760 to "reveal" windows lower in the window stack 1760. For example, window 3 1768 is lower in the stack than both window 4 1770 and window 8 1774 but is still displayed. A logical data structure to manage the window stack can be as described in conjunction with Fig. 10.

[0105] When a new window is opened, the newly activated window is generally positioned at the top of the stack. However, where and how the window is positioned within the stack can be

a function of the orientation of the device 100, the context of what programs, functions, software, etc. are being executed on the device 100, how the stack is positioned when the new window is opened, etc. To insert the window in the stack, the position in the stack for the window is determined and the touch sensitive display 110, 114 to which the window is associated may also be determined. With this information, a logical data structure for the window can be created and stored. When user interface or other events or tasks change the arrangement of windows, the window stack(s) can be changed to reflect the change in arrangement. It should be noted that these same concepts described above can be used to manage the one or more desktops for the device 100.

[0106] A logical data structure 1800 for managing the arrangement of windows or desktops in a window stack is shown in Fig. 10. The logical data structure 1800 can be any data structure used to store data whether an object, record, file, etc. The logical data structure 1800 can be stored in any type of database or data storage system, regardless of protocol or standard. In embodiments, the logical data structure 1800 includes one or more portions, fields, attributes, etc. that store data in a logical arrangement that allows for easy storage and retrieval of the information. Hereinafter, these one or more portions, fields, attributes, etc. shall be described simply as fields. The fields can store data for a window identifier 1804, dimensions 1808, a stack position identifier 1812, a display identifier 1816, and/or an active indicator 1820. Each window in a window stack can have an associated logical data structure 1800. While only a single logical data structure 1800 is shown in Fig. 10, there may be more or fewer logical data structures 1800 used with a window stack (based on the number of windows or desktops in the stack), as represented by ellipses 1824. Further, there may be more or fewer fields than those shown in Fig. 10, as represented by ellipses 1828.

[0107] A window identifier 1804 can include any identifier (ID) that uniquely identifies the associated window in relation to other windows in the window stack. The window identifier 1804 can be a globally unique identifier (GUID), a numeric ID, an alphanumeric ID, or other type of identifier. In embodiments, the window identifier 1804 can be one, two, or any number of digits based on the number of windows that can be opened. In alternative embodiments, the size of the window identifier 1804 may change based on the number of windows opened. While the window is open, the window identifier 1804 may be static and remain unchanged.

[0108] Dimensions 1808 can include dimensions for a window in the composite display 1760. For example, the dimensions 1808 can include coordinates for two or more corners of the window or may include one coordinate and dimensions for the width and height of the window. These dimensions 1808 can delineate what portion of the composite display 1760 the window may occupy, which may the entire composite display 1760 or only part of composite display 1760. For example, window 4 1770 may have dimensions 1880 that indicate that the window 1770 will occupy only part of the display area for composite display 1760, as shown in Figs. 9C through 9E. As windows are moved or inserted in the window stack, the dimensions 1808 may change.

[0109] A stack position identifier 1812 can be any identifier that can identify the position in the stack for the window or may be inferred from the window's control record within a data structure, such as a list or a stack. The stack position identifier 1812 can be a GUID, a numeric ID, an alphanumeric ID, or other type of identifier. Each window or desktop can include a stack position identifier 1812. For example, as shown in Fig. 9A, window 1 1704 in stack 1 1760 can have a stack position identifier 1812 of 1 identifying that window 1704 is the first window in the stack 1760 and the active window. Similarly, window 6 1724 can have a stack position identifier 1812 of 3 representing that window 1724 is the third window in the stack 1760. Window 2 1708 can also have a stack position identifier 1812 of 1 representing that window 1708 is the first window in the second stack 1764. As shown in Fig. 9B, window 1 1744 can have a stack position identifier 1812 of 1, window 3, rendered in portions 1732 and 1736, can have a stack position identifier 1812 of 3, and window 6 1756 can have a stack position identifier 1812 of 6. Thus, depending on the type of stack, the stack position identifier 1812 can represent a window's location in the stack.

[0110] A display identifier 1816 can identify that the window or desktop is associated with a particular display, such as the first display 110 or the second display 114, or the composite display 1760 composed of both displays. While this display identifier 1816 may not be needed for a multi-stack system, as shown in Fig. 9A, the display identifier 1816 can indicate whether a window in the serial stack of Fig. 9B is displayed on a particular display. Thus, window 3 may have two portions 1732 and 1736 in Fig. 9B. The first portion 1732 may have a display identifier 1816 for the first display while the second portion 1736 may have a display identifier 1816 for the second display 114. However, in alternative embodiments, the window may have

two display identifier 1816 that represent that the window is displayed on both of the displays 110, 114, or a display identifier 1816 identifying the composite display. In another alternate embodiment, the window may have a single display identifier 1816 to represent that the window is displayed on both of the displays 110, 114.

[0111] Similar to the display identifier 1816, an active indicator 1820 may not be needed with the dual stack system of Fig. 9A, as the window in stack position 1 is active and displayed. In the system of Fig. 9B, the active indicator 1820 can indicate which window(s) in the stack is being displayed. Thus, window 3 may have two portions 1732 and 1736 in Fig. 9. The first portion 1732 may have an active indicator 1820 while the second portion 1736 may also have an active indicator 1820. However, in alternative embodiments, window 3 may have a single active indicator 1820. The active indicator 1820 can be a simple flag or bit that represents that the window is active or displayed.

[0112] An embodiment of a method 900 for creating a window stack is shown in Fig. 11. While a general order for the steps of the method 1900 is shown in Fig. 11. Generally, the method 1900 starts with a start operation 1904 and ends with an end operation 1928. The method 1900 can include more or fewer steps or can arrange the order of the steps differently than those shown in Fig. 11. The method 1900 can be executed as a set of computer-executable instructions executed by a computer system and encoded or stored on a computer readable medium. Hereinafrer, the method 1900 shall be explained with reference to the systems, components, modules, software, data structures, user interfaces, etc. described in conjunction with Figs. 1-10.

[0113] A multi-screen device 100 can receive activation of a window, in step 1908. In embodiments, the multi-screen device 100 can receive activation of a window by receiving an input from the touch sensitive display 110 or 114, the configurable area 112 or 116, a gesture capture region 120 or 124, or some other hardware sensor operable to receive user interface inputs. The processor may execute the Task Management Module 540 may receive the input. The Task Management Module 540 can interpret the input as requesting an application task to be executed that will open a window in the window stack.

[0114] In embodiments, the Task Management Module 540 places the user interface interaction in the task stack 552 to be acted upon by the Display Configuration Module 568 of the Multi-Display Management Module 524. Further, the Task Management Module 540 waits

for information from the Multi-Display Management Module 524 to send instructions to the Window Management Module 532 to create the window in the window stack.

[0115] The Multi-Display Management Module 524, upon receiving instruction from the Task Management Module 540, determines to which touch portion of the composite display 1760, the newly activated window should be associated, in step 1912. For example, window 4 1770 is associated with the a portion of the composite display 1764 In embodiments, the device state module 574 of the Multi-Display Management Module 524 may determine how the device is oriented or in what state the device is in, e.g., open, closed, portrait, etc. Further, the preferences module 572 and/or requirements module 580 may determine how the window is to be displayed. The gesture module 576 may determine the user's intentions about how the window is to be opened based on the type of gesture and the location of where the gesture is made.

[0116] The Display Configuration Module 568 may use the input from these modules and evaluate the current window stack 1760 to determine the best place and the best dimensions, based on a visibility algorithm, to open the window. Thus, the Display Configuration Module 568 determines the best place to put the window at the top of the window stack 1760, in step 1916. The visibility algorithm, in embodiments, determines for all portions of the composite display, which windows are at the top of the stack. For example, the visibility algorithm determines that window 3 1768, window 4 1770, and window 8 1774 are at the top of the stack 1760 as viewed in Figs. 9C through 9E. Upon determining where to open the window, the Display Configuration Module 568 can assign a display identifier 816 and possibly dimensions 808 to the window. The display identifier 816 and dimensions 808 can then be sent back to the Task Management Module 540. The Task Management Module 540 may then assign the window a stack position identifier 812 indicating the windows position at the top of the window stack.

[0117] In embodiments, the Task Management Module 540 sends the window stack information and instructions to render the window to the Window Management Module 532. The Window Management Module 532 and the Task Management Module 540 can create the logical data structure 800, in step 1924. Both the Task Management Module 540 and the Window Management Module 532 may create and manage copies of the window stack. These copies of the window stack can be synchronized or kept similar through communications between the Window Management Module 532 and the Task Management Module 540. Thus,

the Window Management Module 532 and the Task Management Module 540, based on the information determined by the Multi-Display Management Module 524, can assign dimensions 808, a stack position identifier 812 (e.g., window 1 1782, window 4 1770, etc.), a display identifier 816 (e.g., touch sensitive display 1 110, touch sensitive display 2 114, composite display identifier, etc,), and an active indicator 820, which is generally always set when the window is at the "top" of the stack. The logical data structure 800 may then be stored by both the Window Management Module 532 and the Task Management Module 540. Further, the Window Management Module 532 and the Task Management Module 540may thereinafrer manage the window stack and the logical data structure(s) 800.

[0118] FIG. 12 depicts a further window stacking configuration. A plurality of windows 1, 2, 3, 4, 5, 6, 7, and 8, whether from the same or different multi-screen or single-screen applications, are depicted. The touch sensitive display 110 currently has window 4 in the active display position while touch sensitive display 114 currently has window 5 in the active display position. The stack for touch sensitive display 110, from top to bottom, has window 4 in the active display position and windows 3, 2, and 1 positioned behind it. The stack for touch sensitive display 114, from top to bottom, has window 5 in the active display position and windows 6, 7, and 8 positioned behind it.

[0119] The desktops D1, D2, D3, D4, D5, and D6 are positioned behind the window stacks. The desktops can be viewed as a desktop stack different from the window stack. Viewed in this manner, the touch sensitive display 110 has a corresponding desktop stack comprising desktops D3, D2, and D1,with desktop D1 being at the bottom 2300 stack position and with desktop D3 at the top stack position being capable of being displayed with window 4 (depending on the window position and size (whether maximized or minimized), and the touch sensitive display 114 has a corresponding desktop stack has a corresponding desktop stack comprising desktops D4, D5, and D6,with desktop D6 being at the bottom 2304 stack position and with desktop D4 at the top stack position being capable of being displayed with window 5 (depending on the window position and size (whether maximized or minimized). Conceptually, the desktops can be viewed as a canvas divided, in this example, into six sections of which two can be displayed, at any one time, the touch sensitive displays 110, 114. When the device 100 is in the closed state, this conceptual model, in one configuration, persists. In this configuration, only one window and desktop stacks can be seen (that corresponding to the primary screen) but the other

window and desktop stacks are virtual; that is, they are maintained in memory but cannot be seen because the secondary screen is not enabled.

[0120] Single-screen applications are typically related only to one window while multi-screen applications can be related to one or more windows. The latter hierarchical relationship is depicted in Fig. 8. As shown in Fig. 8, first window 1.1 is a root window, second window 1.1.1 is a node window, and third window 1.1.1.1 is a leaf window. The windows are in a hierarchical tree-like sequence and are controlled by a common multi-screen application. The root window 1.1 may be a top level view of the hierarchical application window sequence such that there is no parent window corresponding to the root window. The root window may be a parent to a node window. One or more node windows may be provided that are related as parent/children. A node window may also serve as a leaf window. By leaf window, it is meant that the leaf window has no corresponding node screen for which the leaf window is a parent. As such, the leaf window does not have any children node windows, Windows 1.1.1.1 and 1.1.2 are examples of leaf windows. The root view refers to all of the windows controlled by a multi-screen application at a point in time.

[0121] A number of examples will now be discussed with reference to minimizing and maximizing root view windows and subcomponents thereof. Prior to discussing the examples, some fundamental rules regarding maximizing and minimizing displayed images will be discussed. A maximize or minimize operation is typically initiated by a user gesture, particularly a flip or drag. In a drag gesture, a 1:1 map reaction between the user's digit and the displayed image is used as the user "drags" the displayed image to the other touch sensitive display. The user typically releases the displayed image before it is fully moved to the other touch sensitive display. The initial speed after release is generally decreased progressively or gradually until the displayed image is fully located in the desired position. Too little displacement of the displayed image generally will not cause the image to be relocated. Such a low level of displacement is assumed to be accidental or incidental to the user using the device.

[0122] In a root view, or set of hierarchically related windows, the user, when the device is in dual screen mode, can maximize the related windows of the multi-screen application to enable side-by-side viewing of the related windows (e.g., the parent and child windows), thereby viewing the window hierarchy as a linear progression, with the top-most window being displayed on the lefr and the lowest-most or deepest level window being displayed on the right (or vice

versa). As discussed below, this visual configuration can be controlled using directional transitions conveying a navigation direction and location of a window in relation to a launching point of the window.

[0123] A multi-screen application window generally opens in minimized mode. A continuing drag and release gesture when a window of a multi-screen application, originally minimized in a first touch sensitive display, is at least half-way across the second touch sensitive display will generally cause the window to be maximized, or displayed in both the first and second touch sensitive displays. Dragging a window of a multi-screen application originally minimized in a first touch sensitive display to a second touch sensitive display all the way across to the far edge of the second touch sensitive display will generally maximize the window across both the first and second touch sensitive displays. When the user continues to drag in the same direction after the window of the multi-screen application is expanded fully (or maximized across both the first and second touch sensitive displays and touching both far (non-adjacent) edges of the first and second touch sensitive displays), the window will pull away from the far edge of the first touch sensitive display and, once the far edge of the window is half-way across the first touch sensitive display, the application window will generally reposition in minimized mode in the second touch sensitive display.

[0124] The flick gesture will not generally maximize a window of a multi-screen application but generally will reposition a minimized window of a single- or multi-screen application. The flick gesture will generally not minimize window(s) of a maximized application. The flick gesture is a way to move rapidly through the windows of a stack, whether the windows are of a single- or multi-screen application.

[0125] For instance, a user selects an application shortcut on a first desktop D1 to launch a multi-screen application, such as Outlook™. The application opens to display a defined parent or root window 1.1 in the screen of the first desktop D1 (or the screen from which the application was launched). The user expands or maximizes the application to both the first and second touch sensitive displays to display the parent window 1.1 and the node window 1.1.1. Once expanded, the node window 1.1.1 appears based on rules set by the particular application, e.g., the node window 1.1.1 selected is at a selected position and/or relationship to the parent window 1.1. The user selects, such as by a tap gesture, the node window 1.1.1 towards the other first touch sensitive display (displaying the parent window 1.1, causing the parent window 1.1 to slide off-

display and the leaf window 1.1.1.1 to appear in the second touch sensitive display. If, on the other hand, the user selects, such as by a tap gesture, the parent window 1.1 moves to and is displayed by the second touch sensitive display, the node window 1.1.1 slides to the right off the second touch sensitive display to an inactive display position, and a window 1 (not shown) at a hierarchy level higher than the parent window 1.1moves into and is displayed by the first touch sensitive display

[0126] A first example will be applied to Figs. 13A-C. The first and second touch sensitive displays 110, 114 display, respectively, a leaf view window 1.1.1.1 and a second desktop D2 (Fig. 13A). Stated another way, the first touch sensitive display 110 has the leaf view window 1.1.1.1 (1308) in an active display position, and the second touch sensitive display 114 has the second desktop D2 in an active display position. In a gesture capture region 120 or 124 (or in other configurations the first or second touch sensitive display 110 or 114) receives, from the user, a gesture 1300. Exemplary gestures include a drag or flick to the right(though one or more other gestures may be employed). By the gesture 1300, the user seeks to maximize the root view to show the node window 1.1.1 (1304) and the leaf window 1.1.1.1 (1308) in separate displays (Fig. 13B). As a general rule, a leaf window (or window having no child) is positioned on the second touch sensitive display 114. Thus, the leaf window 1.1.1.1 (1308) moves to the active display position on the second touch sensitive display 114, and the node window 1.1.1 (1304) is moved to the active display position on the first touch sensitive display 110 as shown by Fig. 13C. The second desktop D2, meanwhile, moves to an inactive display position on the second touch sensitive display 114. The leaf window 1.1.1.1 (1308) does this by sliding over to the second touch sensitive display 114, thereby revealing or uncovering the node window 1.1.1 (1304). As will be appreciated, a reverse general rule could also be applied; that is, the leaf window could be positioned on the first touch sensitive display 110.

[0127] A second example will be discussed with reference to Figs. 14A-C in which the first touch sensitive display 110 has a first desktop D1 in an active display position, and the second touch sensitive display 114 has the leaf view window 1.1.1.1 (1308) in an active display position. In a gesture capture region 120 or 124 (or in other configurations the first or second touch sensitive display 110 or 114) receives, from the user, a gesture 1300, such as a flick or drag, to the left. By the gesture 1300, the user seeks to maximize the root view to show the node window 1.1.1 (1304) and the leaf window 1.1.1.1 (1308) in separate displays (Fig. 14B). As a general

rule, a leaf window (or window having no child) is positioned on the second touch sensitive display 114. Thus, the leaf window 1.1.1.1 (1308) remains in the active display position on the second touch sensitive display 114, and the node window 1.1.1 (1304) is moved to the active display position on the first touch sensitive display 110 as shown by Fig. 14C. The first desktop D1, meanwhile, moves to an inactive display position on the first touch sensitive display 110. The node window 1.1.1 (1304) does this by sliding from under the leaf window 1.1.1.1 (1308). As will be appreciated, a reverse general rule could also be applied; that is, the leaf window could be positioned on the first touch sensitive display 110.

[0128] A third example will be discussed with reference to Figs. 24A-C in which the first touch sensitive display 110 has the root window 1.1 (1600) in an active display position (and the node and leaf windows 1.1.1 (1304) and 1.1.1.1 (1308) in inactive display positions), and the second touch sensitive display 114 has the second desktop D2 in an active display position. In a gesture capture region 120 or 124 (or in other configurations the first or second touch sensitive display 110 or 114) receives, from the user, a gesture 1300, such as a flick or drag, to the right. By the gesture 1300, the user seeks to maximize the root view to show the root window 1.1 (1600) and node window 1.1.1 (1304) in separate displays (Fig. 24B). As a general rule, whenever a child window (e.g., a node or leaf window) is uncovered through maximization, it is uncovered on the second touch sensitive display 114. Thus, the root window 1.1 (1600) remains in the active display position on the first touch sensitive display 110, and the node window 1.1.1 (1304) is moved to the active display position on the second touch sensitive display 114 as shown by Fig. 24C. The second desktop D2, meanwhile, moves to an inactive display position on the second touch sensitive display 114. The node window 1.1.1 (1304) does this by sliding out from under the root window 1.1 (1600). As will be appreciated, a reverse general rule could also be applied; that is, the child window could be positioned on the first touch sensitive display 110.

[0129] A fourth example will be discussed with reference to Figs. 25A-C in which the first touch sensitive display 110 has the first desktop D1 in an active display position, and the second touch sensitive display 114 has the root window 1.1 (1600) in an active display position (and the node and leaf windows 1.1.1 (1304) and 1.1.1.1 (1308) in inactive display positions). In a gesture capture region 120 or 124 (or in other configurations the first or second touch sensitive display 110 or 114) receives, from the user, a gesture 1300, such as a flick or drag, to the left.

By the gesture 1300, the user seeks to maximize the root view to show the root window 1.1 (1600) and node window 1.1.1 (1304) in separate displays (Fig. 25B). Thus, the node window 1.1.1 (1304) moves to the active display position on the second touch sensitive display 114, and the root window 1.1 (1600) is moved to the active display position on the first touch sensitive display 110 as shown by Fig. 25C. The first desktop D1, meanwhile, moves to an inactive display position on the first touch sensitive display 110. The root window 1.1 (1600) does this by sliding over from above the node window 1.1.1 (1304).

[0130] A fifth example will be discussed with reference to Figs. 15A-C in which the first touch sensitive display 110 displays the leaf window 1.1.1.1 (1308), and the second touch sensitive display 114 displays the second desktop D2. In a gesture capture region 120 or 124 (or in other configurations the first or second touch sensitive display 110 or 114) receives, from the user, a gesture 1300, particularly a flick gesture, to the right. By the gesture 1300, the user seeks to move the leaf window 1.1.1.1 (1308) to the second touch sensitive display 114 to reveal the first desktop D1. As a general rule, a flick gesture will move the leaf window to the other touch sensitive display. Thus, the leaf window 1.1.1.1 (1308) moves to the active display position on the second touch sensitive display 114, and the first desktop D1 moves to the active display position on the first touch sensitive display 110 as shown by Fig. 15C. The second desktop D2, meanwhile, moves to an inactive display position on the second touch sensitive display 110.

[0131] A sixth example will be discussed with reference to Figs. 26A-C in which the first touch sensitive display 110 displays the root window 1.1 (1600), and the second touch sensitive display 114 displays the second desktop D2. As will be appreciated, the display of a higher level (e.g., node) window by the device 100 indicates that its related, lower level windows are positioned in the same display "underneath" the higher level window (or in inactive display positions in the stack corresponding to the appropriate display). With reference to Fig. 26A, the display of the root window 1.1 (1600) indicates that the node window 1.1.1 (1304) and leaf window 1.1.1.1 (1600) are in inactive display positions in the first touch sensitive display 110. In a gesture capture region 120 or 124 (or in other configurations the first or second touch sensitive display 110 or 114) receives, from the user, a gesture 1300, particularly a flick gesture, to the right. By the gesture 1300, the user seeks to move the root view to the second touch sensitive display 114 to reveal the first desktop D1. As a general rule, a flick gesture will move the root window (or root view) to the other touch sensitive display (and maintain the relative

stack display positions of the root view). Thus, the root window 1.1 (1600) moves to the active display position on the second touch sensitive display 114, and the first desktop D1 moves to the active display position on the first touch sensitive display 110 as shown by Fig. 26C. The second desktop D2, meanwhile, moves to an inactive display position on the second touch sensitive display 110.

[0132] A seventh example will be applied to Figs. 16A-C. The first and second touch sensitive displays 110, 114 display, respectively, a root window 1.1 (1600) and a node window 1.1.1 (1304) (Fig. 16A). Stated another way, the first touch sensitive display 110 has the root window 1.1 (1600) in an active display position, and the second touch sensitive display 114 has the node window 1.1.1 (1304) in an active display position. In a gesture capture region 120 or 124 (or in other configurations the first or second touch sensitive display 110 or 114) receives, from the user, a gesture 1300. Exemplary gestures include a drag or flick to the right. By the gesture 1300, the user seeks to minimize the root view in the second touch sensitive display 114 (Fig. 16B). As a general rule, a minimizing a higher level (e.g., root) and lower level (e.g., node) windows from a touch sensitive display is done so as to maintain the higher level (e.g., root or parent) window 1.1 (1600) in the active display position on the minimized view. Thus, the root window 1.1 (1600) moves to the active display position on the second touch sensitive display 114, and the node window 1.1.1 (1304) is moved to the inactive display position on the second touch sensitive display 114 as shown by Fig. 16C. The first desktop D1, meanwhile, moves to the active display position on the first touch sensitive display 110. The node window 1.1 (1600) does this by sliding over to the second touch sensitive display 114, thereby revealing or uncovering the first desktop D1. As will be appreciated, a reverse general rule could also be applied; that is, the node window could be maintained in view over the root window.

[0133] An eighth example will be discussed with reference to Figs. 17A-C in which the first touch sensitive display 110 has the root window 1.1 (1600) in an active display position, and the second touch sensitive display 114 has the node window 1.1.1 (1304) in an active display position. In a gesture capture region 120 or 124 (or in other configurations the first or second touch sensitive display 110 or 114) receives, from the user, a gesture 1300, such as a flick or drag, to the lefr. By the gesture 1300, the user seeks to minimize the root view in the first touch sensitive display 110 (Fig. 17B). The root window 1.1 (1600) remains in the active display position on the first touch sensitive display 110, and the node window 1.1.1 (1304) is moved to

an inactive display position on the first touch sensitive display 110 as shown by Fig. 17C. The second desktop D2, meanwhile, moves to an active display position on the second touch sensitive display 114. The node window 1.1.1 (1304) does this by sliding under the root window 1.1 (1600).

[0134] A ninth example will be discussed with reference to Figs. 18A-C in which the first touch sensitive display 110 displays the node window 1.1.1 (1304), and the second touch sensitive display 114 displays the leaf window 1.1.1.1 (1308). In a gesture capture region 120 or 124 (or in other configurations the first or second touch sensitive display 110 or 114) receives, from the user, a gesture 1300, particularly a flick or drag gesture, to the right. By the gesture 1300, the user seeks to move the node window 1.1.1 (1304) to the second touch sensitive display 114 to reveal the first desktop D1 in the first touch sensitive display 110. Thus, the leaf window 1.1.1.1 (1308) moves to an inactive display position on the second touch sensitive display 114, and the first desktop D1 moves to the active display position on the first touch sensitive display 110 as shown by Fig. 18C. The node window 1.1.1 (1304), meanwhile, moves (or slides over the leaf window 1.1.1.1 (1308) to the active display position on the second touch sensitive display 114.

[0135] A tenth example will be discussed with reference to Figs. 19A-C in which the first touch sensitive display 110 displays the node window 1.1.1 (1304), and the second touch sensitive display 114 displays the leaf window 1.1.1.1 (1308). In a gesture capture region 120 or 124 (or in other configurations the first or second touch sensitive display 110 or 114) receives, from the user, a gesture 1300, particularly a flick or drag gesture, to the lefr. By the gesture 1300, the user seeks to move the leaf window 1.1.1.1 (1308) to the first touch sensitive display 110 to reveal the second desktop D2 in the second touch sensitive display 114. Thus, the leaf window 1.1.1.1 (1308) moves (or slides under the node window 1.1.1 (1304)) to an inactive display position on the first touch sensitive display 110, and the second desktop D2 moves to the active display position on the second touch sensitive display 114 as shown by Fig. 19C. The node window 1.1.1 (1304), meanwhile, remains in the active display position on the first touch sensitive display 110.

[0136] In a further example shown in Figs. 20A-C, the device 100 is in dual landscape display mode. Unlike dual portrait display mode, the device, in the dual landscape display mode, can

present one view, and not two. This capability, in turn, affects the maximization and minimization rules.

[0137] The first touch sensitive display 110 displays a minimized root window 1.1 (1600), and the second touch sensitive display 114 displays a second desktop D2. The root window 1.1 (1600) is minimized in that it occupies only one of the primary and secondary screens 104 and 108. In a gesture capture region 120 or 124 (or in other configurations the touch sensitive display 110 or 114) receives, from the user, a gesture 1300, such as a flick or drag gesture downwards. By the gesture 1300, the user seeks to maximize the root window 1.1 (1600) by displaying the root window on both the first and second touch sensitive displays 110 and 114. As shown by Fig. 20B, the root view 1.1 slides downwards to cover the second touch sensitive display 114. Thus, the root window 1.1 (1600) is in the active display position in both the first and second touch sensitive displays 110 and 114 (Fig. 20C).

[0138] A further example is shown in Figs. 21A-C. With reference to Fig. 21A, the first touch sensitive display 110 displays a first desktop D1, and the second touch sensitive display 114 displays a minimized root window 1.1 (1600). In a gesture capture region 120 or 124 (or in other configurations the touch sensitive display 110 or 114) receives, from the user, a gesture 1300, such as a flick or drag gesture upwards. By the gesture 1300, the user seeks to maximize the root window 1.1 (1600) by displaying the root window on both the first and second touch sensitive displays 110 and 114. As shown by Fig. 21B, the root view 1.1 slides upwards to cover the second touch sensitive display 114. Thus, the root window 1.1 (1600) is in the active display position in both the first and second touch sensitive displays 110 and 114 (Fig. 21C).

[0139] With reference to Figs. 22A-C, a further example will be discussed. With reference to Fig. 22A, the first touch sensitive display 110 displays the first desktop D1, and the second touch sensitive display 114 displays the root window 1.1 (1600). As will be appreciated, the display of a higher level (e.g., node) window by the device 100 indicates that its related, lower level windows are positioned in the same display "underneath" the higher level window (or in inactive display positions in the stack corresponding to the appropriate display). With reference to Fig. 22A, the display of the root window 1.1 (1600) indicates that the node window 1.1.1 (1304) and leaf window 1.1.1.1 (1600) are in inactive display positions in the second touch sensitive display 110. In a gesture capture region 120 or 124 (or in other configurations the first or second touch sensitive display 110 or 114) receives, from the user, a gesture 1300, particularly a flick gesture,

upwards. By the gesture 1300, the user seeks to move the root view to the first touch sensitive display 110 to reveal the second desktop D2. As a general rule, a flick gesture will move the root window (or root view) to the other touch sensitive display (and maintain the relative stack display positions of the root view). Thus, the root window 1.1 (1600) moves to the active display position on the first touch sensitive display 110, and the second desktop D2 moves to the active display position on the second touch sensitive display 114 as shown by Fig. 22C. The first desktop D1, meanwhile, moves to an inactive display position on the first touch sensitive display 110.

[0140] A further example is shown in Figs. 23A-C. With reference to Fig. 23A, the first and second touch sensitive displays 110 and 114 display a maximized root window 1.1 (1600). In a gesture capture region 120 or 124 (or in other configurations the touch sensitive display 110 or 114) receives, from the user, a gesture 1300, such as a flick or drag gesture upwards. By the gesture 1300, the user seeks to minimize the root window 1.1 (1600) by displaying the root window in the first touch sensitive display 110 to reveal the second desktop D2 in the second touch sensitive display 114. As shown by Fig. 23B, the root view 1.1 in the second touch sensitive display 114 slides upwards, revealing the second desktop D2. Thus, the root window 1.1 (1600) is in the active display position in the first touch sensitive display 110 while the second desktop is in the active display position in the second touch sensitive display 114 (Fig. 23C).

[0141] In the various examples, middleware 520, particularly one or more of the Multi-Display Management (MDM) class 524, a Surface Cache class 528, a Window Management class 532, an Activity Management class 536, and an Application Management class 540, independently or collectively, receive a command to manipulate a displayed set of related windows from a multi-screen application in a predetermined manner, such as by detecting a gesture 1300 (step 2700). In response, middleware 520 determines whether the set of related windows is currently maximized (in which case each of the touch sensitive displays 110 and 114 display one of the related windows) or minimized (in which case only one of the touch sensitive displays one of the windows) (step 2704). In decision diamond 2708, the middleware 520 decides how to process the command based on whether the window(s) is maximized or minimized. When maximized, the middleware 520 determines whether the device 100 is in the portrait or landscape display mode and, when in the landscape display mode, which of the touch sensitive displays 110 and

114 is on top and which is on the bottom (step 2712). When in portrait display mode, the middleware 520 causes the lower level window (e.g., leaf window 1.1.1.1) to be displayed on a predetermined one of the left or right touch sensitive display. When in landscape display mode, the middleware 520 causes the lower level window to be displayed on a predetermined one of the upper or lower touch sensitive display. When the related window is minimized, the middleware 520 determines whether the device 100 is in the portrait or landscape display mode and, when in the landscape display mode, which of the touch sensitive displays 110 and 114 is on top and which is on the bottom (step 2716). When in portrait display mode, the middleware 520 causes the lower level window to no longer be displayed on a predetermined one of the left or right touch sensitive display. When in landscape display mode, the middleware 520 causes the lower level window to no longer be displayed on a predetermined one of the upper or lower touch sensitive display.

[0142] The exemplary systems and methods of this disclosure have been described in relation to communication devices. However, to avoid unnecessarily obscuring the present disclosure, the preceding description omits a number of known structures and devices. This omission is not to be construed as a limitation of the scopes of the claims. Specific details are set forth to provide an understanding of the present disclosure. It should however be appreciated that the present disclosure may be practiced in a variety of ways beyond the specific detail set forth herein.

[0143] Furthermore, while the exemplary aspects, embodiments, and/or configurations illustrated herein show the various components of the system collocated, certain components of the system can be located remotely, at distant portions of a distributed network, such as a LAN and/or the Internet, or within a dedicated system. Thus, it should be appreciated, that the components of the system can be combined in to one or more devices, such as a communication device, or collocated on a particular node of a distributed network, such as an analog and/or digital telecommunications network, a packet-switch network, or a circuit-switched network. It will be appreciated from the preceding description, and for reasons of computational efficiency, that the components of the system can be arranged at any location within a distributed network of components without affecting the operation of the system. For example, the various components can be located in a switch such as a PBX and media server, gateway, in one or more communications devices, at one or more users' premises, or some combination thereof.

Similarly, one or more functional portions of the system could be distributed between a telecommunications device(s) and an associated computing device.

[0144] Furthermore, it should be appreciated that the various links connecting the elements can be wired or wireless links, or any combination thereof, or any other known or later developed element(s) that is capable of supplying and/or communicating data to and from the connected elements. These wired or wireless links can also be secure links and may be capable of communicating encrypted information. Transmission media used as links, for example, can be any suitable carrier for electrical signals, including coaxial cables, copper wire and fiber optics, and may take the form of acoustic or light waves, such as those generated during radio-wave and infra-red data communications.

[0145] Also, while the flowcharts have been discussed and illustrated in relation to a particular sequence of events, it should be appreciated that changes, additions, and omissions to this sequence can occur without materially affecting the operation of the disclosed embodiments, configuration, and aspects.

[0146] A number of variations and modifications of the disclosure can be used. It would be possible to provide for some features of the disclosure without providing others.

[0147] In other embodiments, other rules may be applied. For example, the preferred orientations of the higher and lower level related windows is different from the orientations described above.

[0148] In yet another embodiment, the systems and methods of this disclosure can be implemented in conjunction with a special purpose computer, a programmed microprocessor or microcontroller and peripheral integrated circuit element(s), an ASIC or other integrated circuit, a digital signal processor, a hard-wired electronic or logic circuit such as discrete element circuit, a programmable logic device or gate array such as PLD, PLA, FPGA, PAL, special purpose computer, any comparable means, or the like. In general, any device(s) or means capable of implementing the methodology illustrated herein can be used to implement the various aspects of this disclosure. Exemplary hardware that can be used for the disclosed embodiments, configurations and aspects includes computers, handheld devices, telephones (e.g., cellular, Internet enabled, digital, analog, hybrids, and others), and other hardware known in the art. Some of these devices include processors (e.g., a single or multiple microprocessors), memory, nonvolatile storage, input devices, and output devices. Furthermore, alternative software

implementations including, but not limited to, distributed processing or component/object distributed processing, parallel processing, or virtual machine processing can also be constructed to implement the methods described herein.

[0149] In yet another embodiment, the disclosed methods may be readily implemented in conjunction with software using object or object-oriented software development environments that provide portable source code that can be used on a variety of computer or workstation platforms. Alternatively, the disclosed system may be implemented partially or fully in hardware using standard logic circuits or VLSI design. Whether software or hardware is used to implement the systems in accordance with this disclosure is dependent on the speed and/or efficiency requirements of the system, the particular function, and the particular software or hardware systems or microprocessor or microcomputer systems being utilized.

[0150] In yet another embodiment, the disclosed methods may be partially implemented in software that can be stored on a storage medium, executed on programmed general-purpose computer with the cooperation of a controller and memory, a special purpose computer, a microprocessor, or the like. In these instances, the systems and methods of this disclosure can be implemented as program embedded on personal computer such as an applet, JAVA® or CGI script, as a resource residing on a server or computer workstation, as a routine embedded in a dedicated measurement system, system component, or the like. The system can also be implemented by physically incorporating the system and/or method into a software and/or hardware system.

[0151] Although the present disclosure describes components and functions implemented in the aspects, embodiments, and/or configurations with reference to particular standards and protocols, the aspects, embodiments, and/or configurations are not limited to such standards and protocols. Other similar standards and protocols not mentioned herein are in existence and are considered to be included in the present disclosure. Moreover, the standards and protocols mentioned herein and other similar standards and protocols not mentioned herein are periodically superseded by faster or more effective equivalents having essentially the same functions. Such replacement standards and protocols having the same functions are considered equivalents included in the present disclosure.

[0152] The present disclosure, in various aspects, embodiments, and/or configurations, includes components, methods, processes, systems and/or apparatus substantially as depicted and

described herein, including various aspects, embodiments, configurations embodiments, subcombinations, and/or subsets thereof. Those of skill in the art will understand how to make and use the disclosed aspects, embodiments, and/or configurations after understanding the present disclosure. The present disclosure, in various aspects, embodiments, and/or configurations, includes providing devices and processes in the absence of items not depicted and/or described herein or in various aspects, embodiments, and/or configurations hereof, including in the absence of such items as may have been used in previous devices or processes, e.g., for improving performance, achieving ease and\or reducing cost of implementation.

[0153] The foregoing discussion has been presented for purposes of illustration and description. The foregoing is not intended to limit the disclosure to the form or forms disclosed herein. In the foregoing Detailed Description for example, various features of the disclosure are grouped together in one or more aspects, embodiments, and/or configurations for the purpose of streamlining the disclosure. The features of the aspects, embodiments, and/or configurations of the disclosure may be combined in alternate aspects, embodiments, and/or configurations other than those discussed above. This method of disclosure is not to be interpreted as reflecting an intention that the claims require more features than are expressly recited in each claim. Rather, as the following claims reflect, inventive aspects lie in less than all features of a single foregoing disclosed aspect, embodiment, and/or configuration. Thus, the following claims are hereby incorporated into this Detailed Description, with each claim standing on its own as a separate preferred embodiment of the disclosure.

[0154] Moreover, though the description has included description of one or more aspects, embodiments, and/or configurations and certain variations and modifications, other variations, combinations, and modifications are within the scope of the disclosure, e.g., as may be within the skill and knowledge of those in the art, after understanding the present disclosure. It is intended to obtain rights which include alternative aspects, embodiments, and/or configurations to the extent permitted, including alternate, interchangeable and/or equivalent structures, functions, ranges or steps to those claimed, whether or not such alternate, interchangeable and/or equivalent structures, functions, ranges or steps are disclosed herein, and without intending to publicly dedicate any patentable subject matter.

What is claimed is:

1. A method, comprising:

receiving, by processor executable middleware in a multi-display communication device, a command to at least one of minimize and maximize a set of related windows from a common multi-screen application, the set of related windows comprising a higher level window and a lower level window at least one of displayed and to be displayed by different touch sensitive displays;

applying, by the processor executable middleware, the following rules:

(B1) when the set of related windows is to be minimized, selecting a touch sensitive display for displaying the higher level window and ceasing to display the lower level window on the other touch sensitive display; and

(B2) when the set of related windows is to be maximized, selecting a predetermined touch sensitive display for displaying one of the higher and lower level windows and causing the other of the higher and lower level windows to be displayed on the other touch sensitive display.

2. The method of claim 1, wherein rule (B2) applies.

3. The method of claim 2, wherein the command is a drag or a flick, wherein neither the higher nor lower level window is displayed in the predetermined touch sensitive display when the gesture is received, and wherein the lower level window appears to move from the other touch sensitive display to the predetermined touch sensitive display, thereby revealing the higher level window in the other touch sensitive display.

4. The method of claim 2, wherein the command is a drag or a flick, wherein the lower level window is in the predetermined touch sensitive display when the gesture is received, wherein the higher level window appears to move from the predetermined touch sensitive display to the other touch sensitive display, whereby the lower level window is displayed by the predetermined touch sensitive display and the higher level window is displayed by the other touch sensitive display.

5. The method of claim 2, wherein the multi-display communication device is in the portrait display mode and wherein the predetermined touch sensitive display is the right-most touch sensitive display.

6. The method of claim 1, wherein rule (B1) applies.

7. The method of claim 6, wherein the command is a drag or a flick, wherein the higher and lower level window are concurrently displayed in the selected and other touch sensitive displays when the gesture is received, wherein the lower level display is displayed in the selected touch sensitive display when the gesture is received, and wherein the higher level window appears to move from the other touch sensitive display to the selected touch sensitive display, thereby covering the lower level window in the other touch sensitive display.

8. The method of claim 6, wherein the command is a drag or a flick, wherein the higher and lower level window are concurrently displayed in the selected and other touch sensitive displays when the gesture is received, wherein the higher level display is displayed in the selected touch sensitive display when the gesture is received, and wherein the lower level window appears to move from the other touch sensitive display to the selected touch sensitive display, thereby disappearing under the higher level window in the other touch sensitive display.

9. A non-transient and tangible computer readable medium comprising processor executable instructions to perform the steps of claim 1.

10. A communication device, comprising:

at least first and second touch sensitive displays to display windows;

a processor; and

a computer readable memory, the computer readable memory comprising processor executable middleware operable to:

receive a command to at least one of minimize and maximize a set of related windows from a common multi-screen application, the set of related windows comprising a higher level window and a lower level window at least one of displayed and to be displayed by different touch sensitive displays;

apply the following rules:

(i) when the set of related windows is to be minimized, selecting a touch sensitive display for displaying the higher level window and ceasing to display the lower level window on the other touch sensitive display; and

(ii) when the set of related windows is to be maximized, selecting a predetermined touch sensitive display for displaying one of the higher and lower level windows and causing the other of the higher and lower level window to be displayed on the other touch sensitive display.

11. The device of claim 10, wherein rule (ii) applies.

12. The device of claim 11, wherein the command is a drag or a flick, wherein neither the higher nor lower level window is displayed in the predetermined touch sensitive display when the gesture is received, and wherein the lower level window appears to move from the other touch sensitive display to the predetermined touch sensitive display, thereby revealing the higher level window in the other touch sensitive display.

13. The device of claim 11, wherein the command is a drag or a flick, wherein the lower level window is in the predetermined touch sensitive display when the gesture is received, wherein the higher level window appears to move from the predetermined touch sensitive display to the other touch sensitive display, whereby the lower level window is displayed by the predetermined touch sensitive display and the higher level window is displayed by the other touch sensitive display.

14. The device of claim 11, wherein the multi-display communication device is in the portrait display mode and wherein the predetermined touch sensitive display is the right-most touch sensitive display.

15. The device of claim 12, wherein rule (ii) applies.

16. The device of claim 15, wherein the command is a drag or a flick, wherein the higher and lower level window are concurrently displayed in the selected and other touch sensitive displays when the gesture is received, wherein the lower level display is displayed in the selected touch sensitive display when the gesture is received, and wherein the higher level window appears to move from the other touch sensitive display to the selected touch sensitive display, thereby covering the lower level window in the other touch sensitive display.

17. The device of claim 15, wherein the command is a drag or a flick, wherein the higher and lower level window are concurrently displayed in the selected and other touch sensitive displays when the gesture is received, wherein the higher level display is displayed in the selected touch sensitive display when the gesture is received, and wherein the lower level window appears to move from the other touch sensitive display to the selected touch sensitive display, thereby disappearing under the higher level window in the other touch sensitive display.

ABSTRACT

[0155] The present disclosure is directed to methodologies and devices for handling maximizing and minimizing of hierarchically related windows.

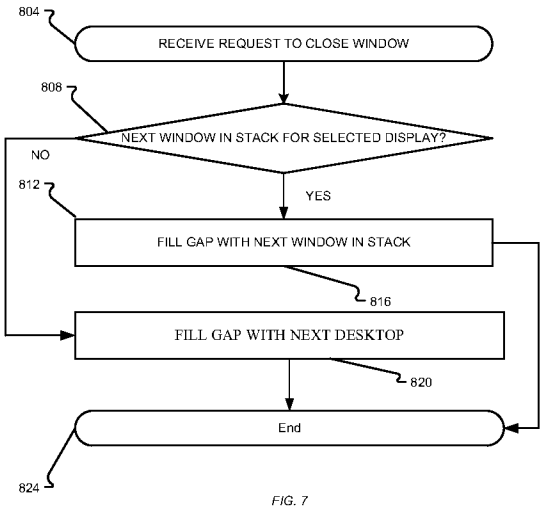


FIG. 7

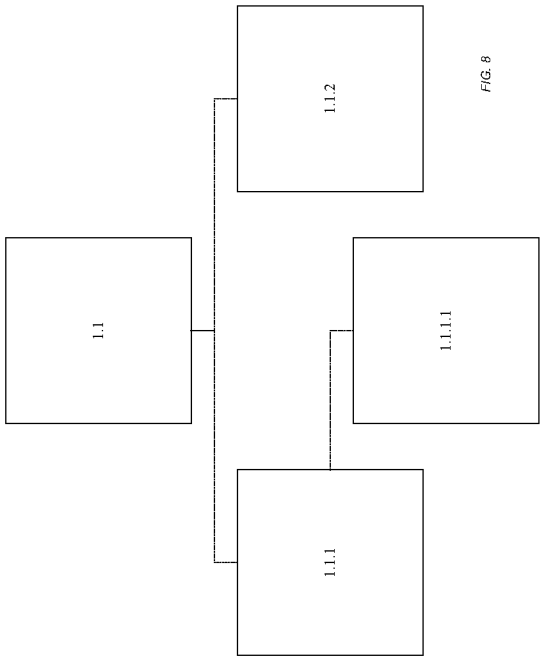


FIG. 8

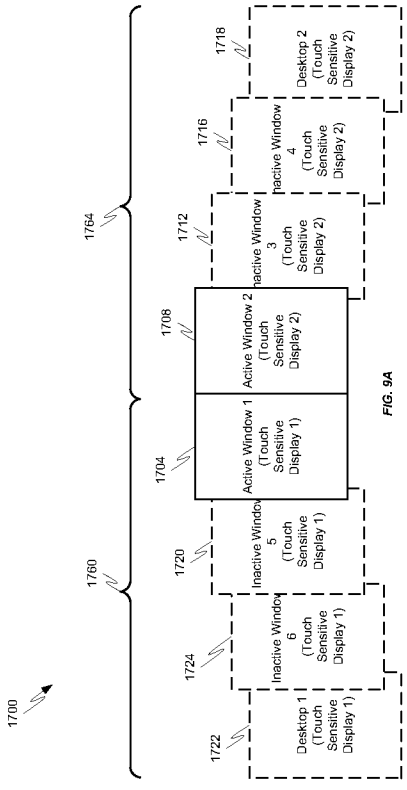


FIG. 9A

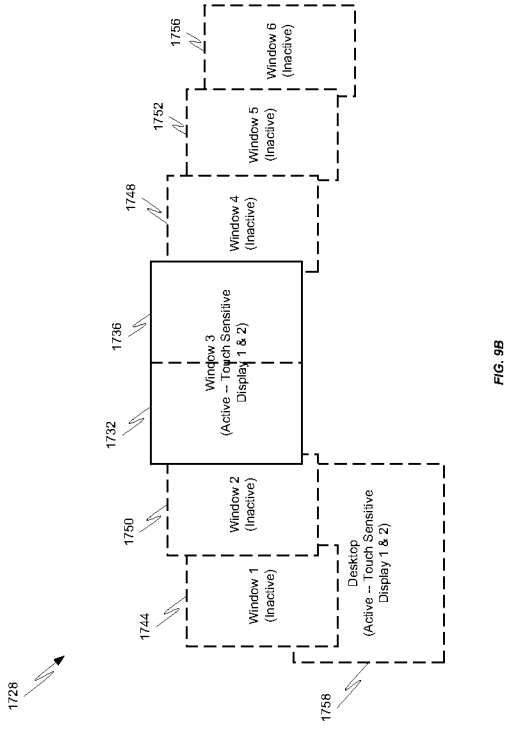


FIG. 9B

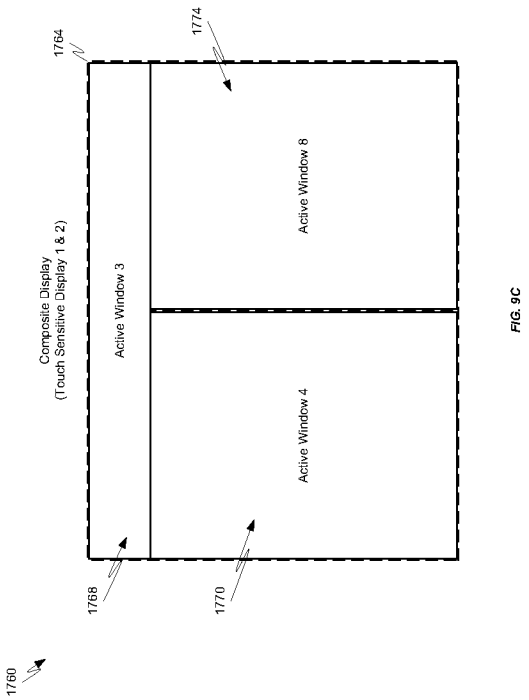


FIG. 9C

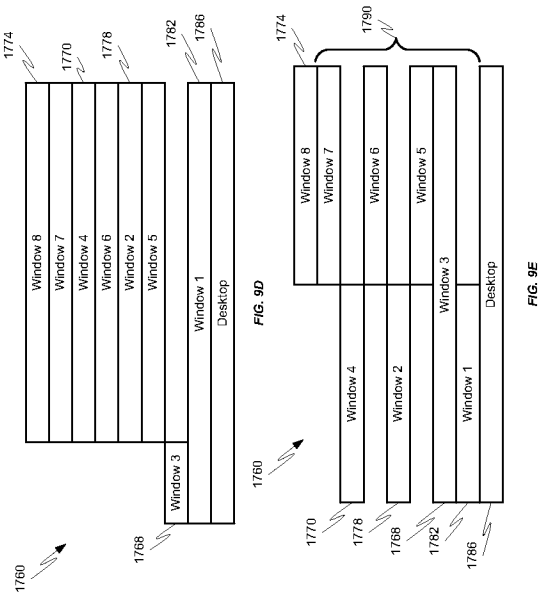


FIG. 9D

FIG. 9E

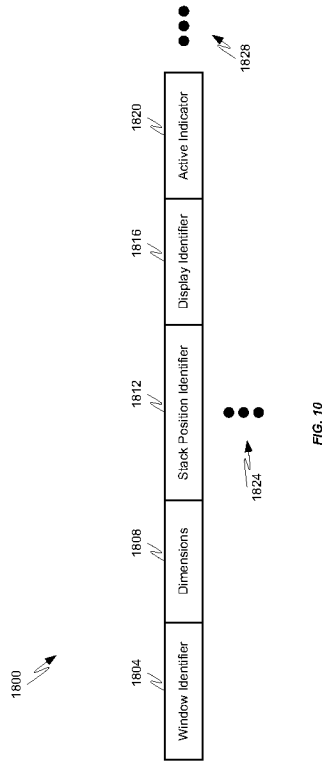


FIG. 10

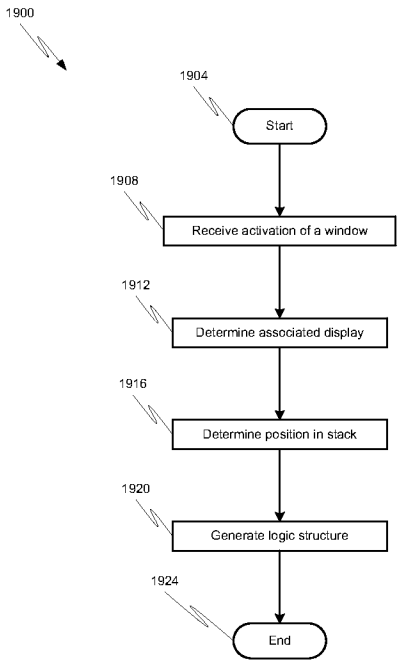


FIG. 11

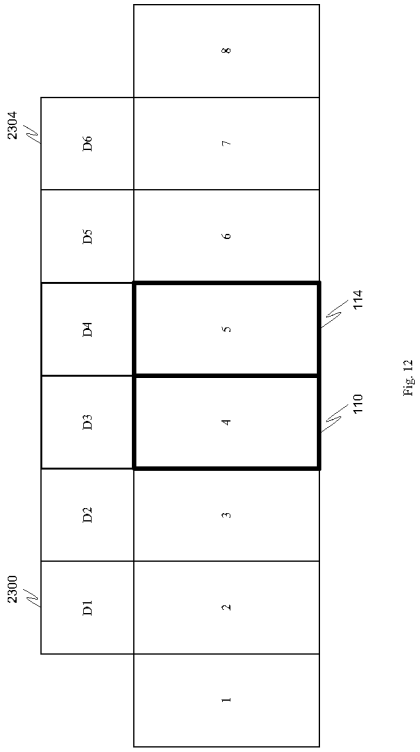
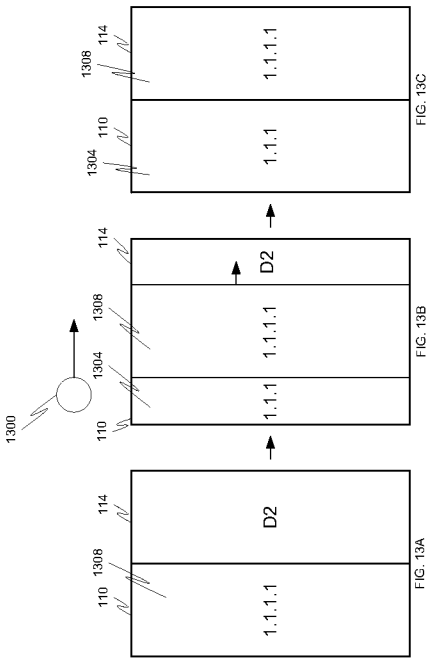
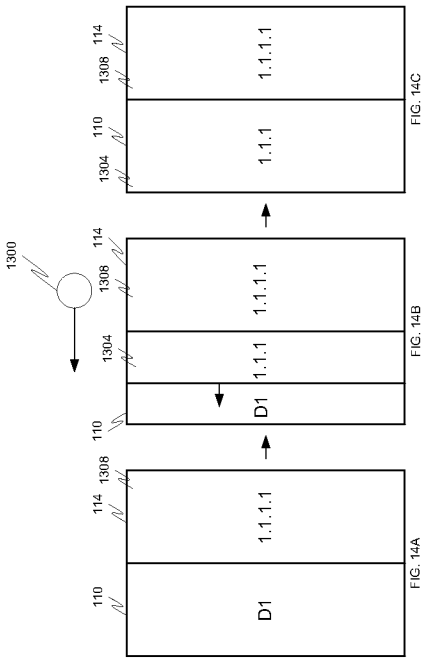
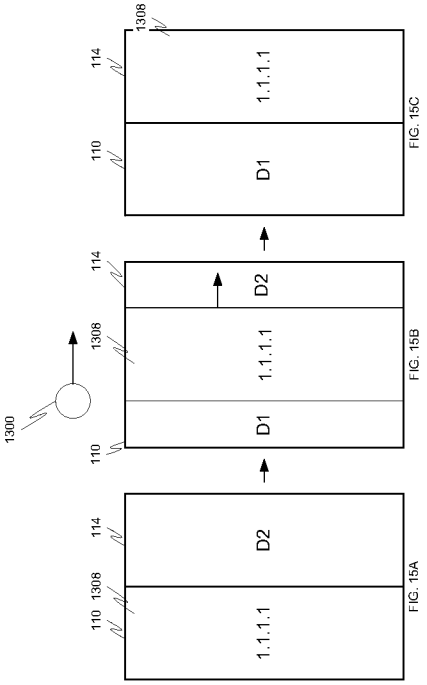
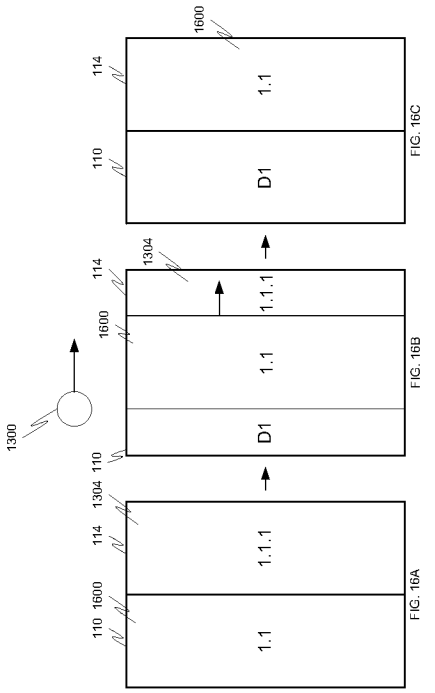

2304
2300
D1
D2
D3
D4
D5
D6
1
2
3
4
5
6
7
8
114
110

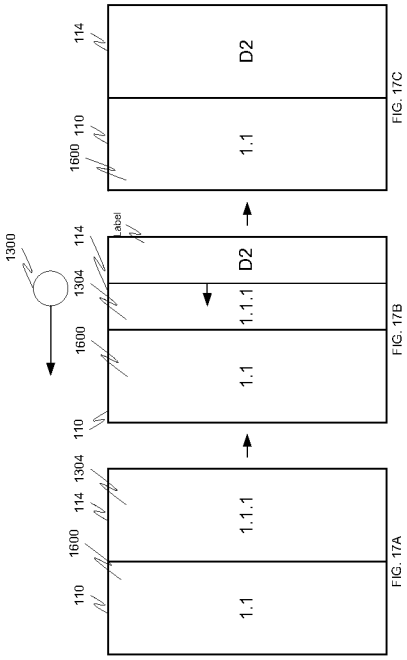
Fig. 12

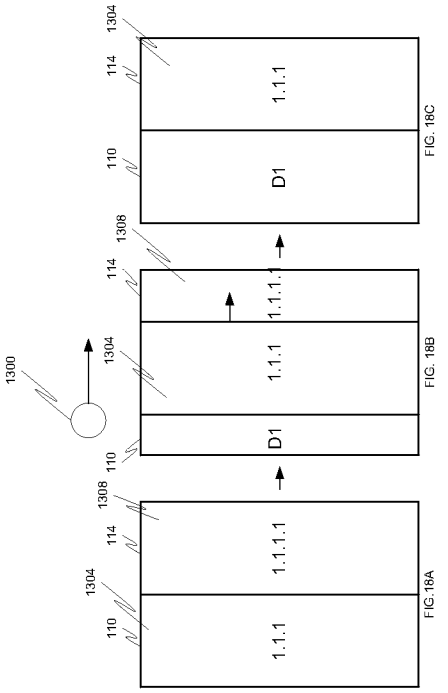

1300
110
1308
114
1304
D2
1.1.1.1
1.1.1
FIG. 13A
FIG. 13B
FIG. 13C

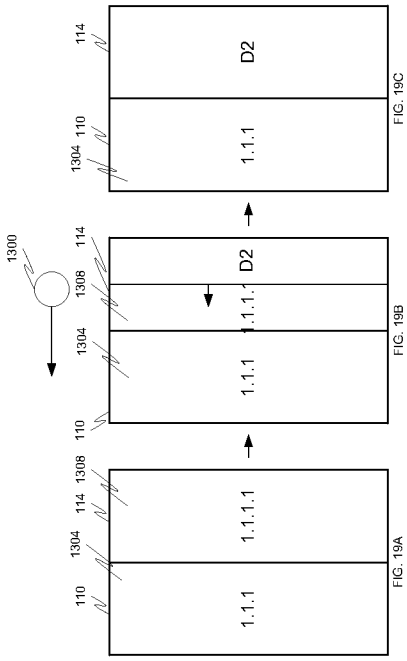

1300
110
1308
114
1304
D1
1.1.1.1
1.1.1
FIG. 14A
FIG. 14B
FIG. 14C


1300
110
1308
114
D1
D2
1.1.1.1
FIG. 15A
FIG. 15B
FIG. 15C


1600
114
110
1.1
D1
FIG. 16C
1300
1304
1.1.1
FIG. 16B
1.1
FIG. 16A


1600
114
110
D2
1.1
FIG. 17C
1300
label
1304
1.1.1
FIG. 17B
FIG. 17A


1304
1308
1.1.1.1
FIG.18A
D1
FIG. 18B
FIG. 18C


D2
1.1.1
FIG. 19C
1300
1308
FIG. 19B
FIG. 19A

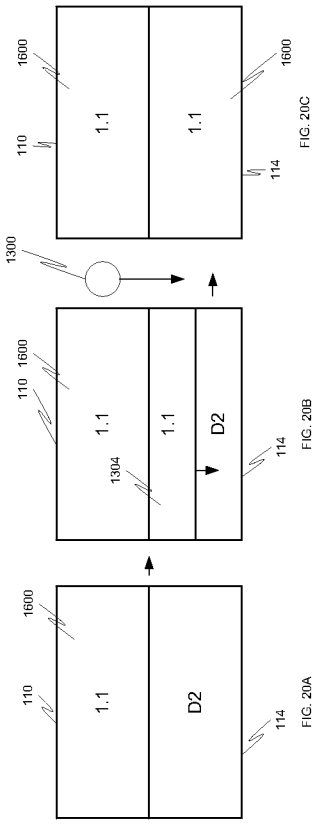


FIG. 20A

FIG. 20B

FIG. 20C

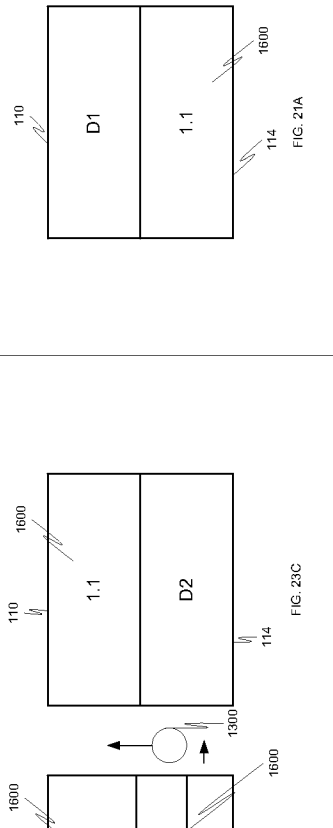


FIG. 21A

FIG. 21B

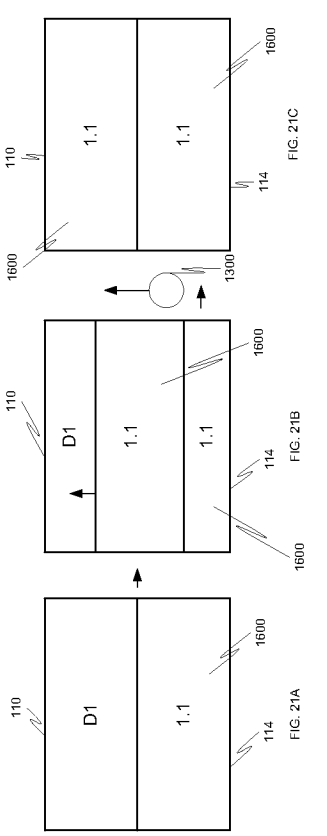


FIG. 21C

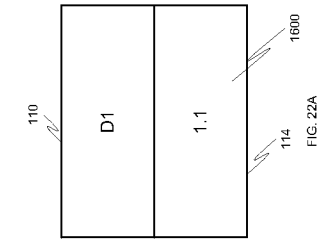


FIG. 22A

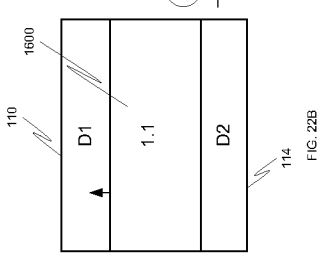


FIG. 22B

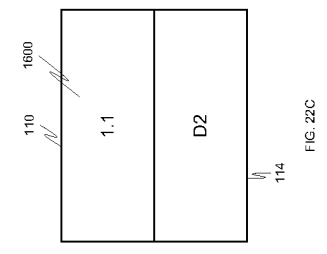


FIG. 22C

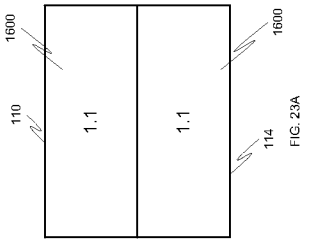


FIG. 23A

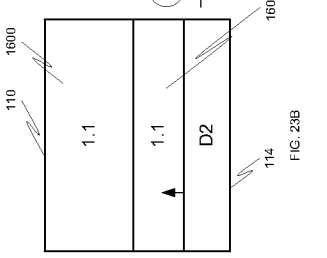


FIG. 23B

FIG. 23C

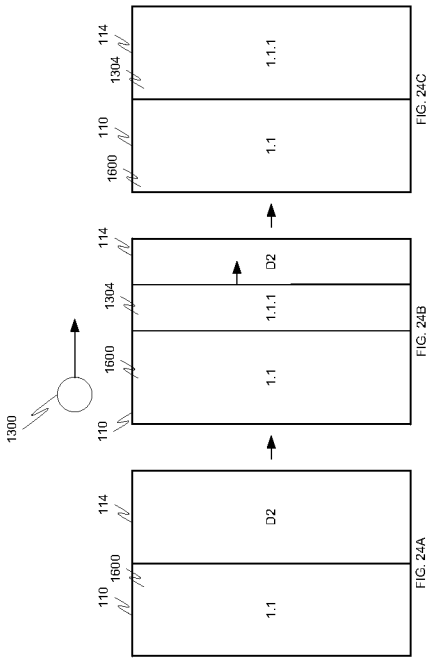


FIG. 24C

FIG. 24B

FIG. 24A

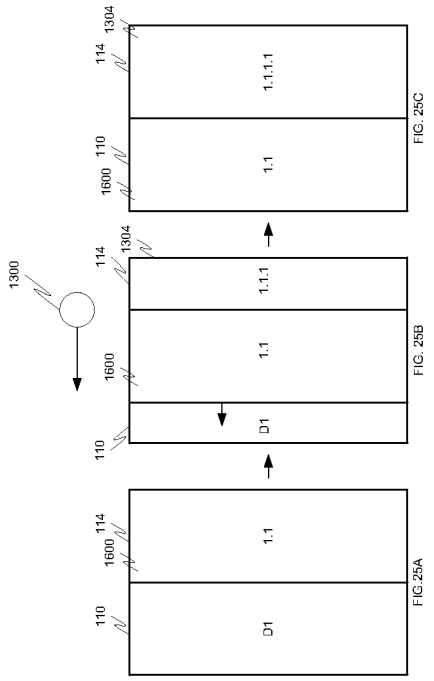


FIG. 25C

FIG. 25B

FIG. 25A

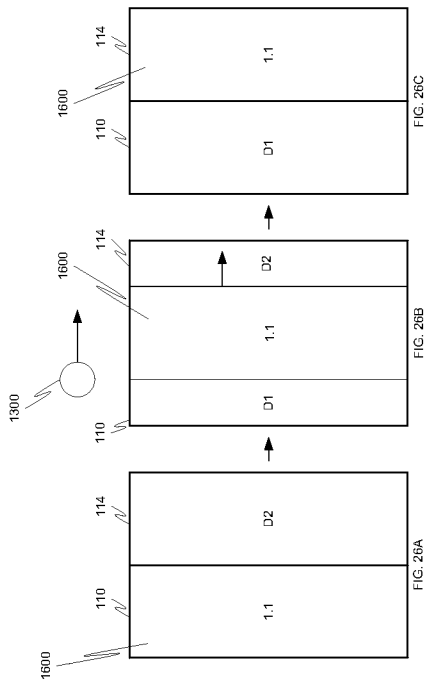


FIG. 26C

FIG. 26B

FIG. 26A

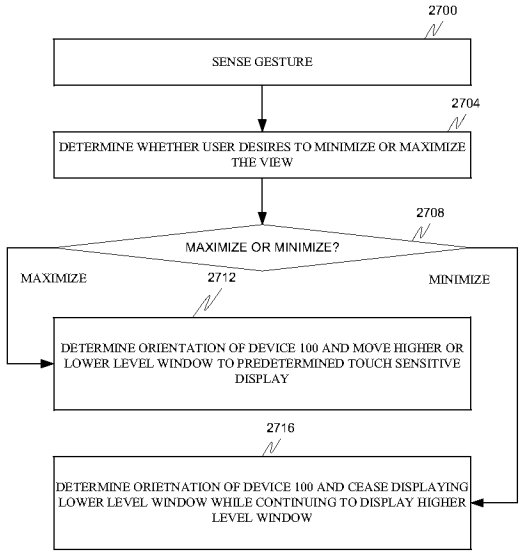


FIG. 27

MULTI-SCREEN DISPLAY CONTROL

[0001] Display Controls:

[0002] Figs. 7A thru 7E depict five examples of gesture sequences that can cause the device 100 to alter the display of data thereon. Specifically, Fig. 7A depicts a first gesture sequence 700 which results in a first display action. Although the first display action is depicted as being performed on the touch sensitive displays 110, 114, it should be appreciated that the first display action can alternatively, or additionally, be performed on other parts of the primary screen 104 and/or secondary screen 108. The first gesture sequence 700 begins when the device 100 is in an open landscape state 348. While in this state, the device 100 displays data on both its displays 110, 114. In other words, both the primary screen 104 and secondary screen 108 are active and pixel data is presented on both displays 110, 114. In some embodiments, the device 100 may display application data from a first application 1.1 on an upper display (e.g., display 110) and not display data from the first application 1.1 on a lower display (e.g., display 114). The data from the first application 1.1 may be displayed on the upper display in a first landscape configuration 612 according to minimization display rules determined by the MDM module 524. The display 114 not currently displaying data from the first application 1.1 may be displaying data from another application (e.g., application 2.1, 3.1, etc.) or it may be displaying desktop data.

[0003] In some embodiments, desktop data may be stacked into a plurality of desktop displays (e.g., a first desktop display D1, a second desktop display D2, a third desktop display D3, an nth desktop display Dn, etc.) and the desktop displays may be ordered much like a stack of cards such that if two screens are currently active and displaying desktop data, then one of the cards (e.g., a first desktop display D1) from the stack of cards is visible on one of the screens and the next card (e.g., a second desktop display D2) from the stack of cards is visible on the other of the screens. All other desktop displays may not currently be visible, but the order of the desktop displays may be maintained in memory by the MDM module 524 or some other display-controlling module. This ordering of desktop displays may be referred to as a desktop stack and the order of desktop displays may be referenced by the MDM module 524 to determine which desktop displays should be presented on which displays 110, 114 in various circumstances. In the example depicted in Fig. 7A, a second desktop display D2 may be initially displayed on the

[0009] The transition area 708 may progressively grow across the lower display 114 until data from the first application 1.1 being maximized fills both displays 110, 114. Until the transition area 708 has completely filled the lower display 114, the data previously presented on the lower display 114 (e.g., the second desktop display D2) may be presented in the remaining portion of the display 114 not occupied by the transition area 708.

[0010] In some embodiments, after the transition area 708 has completely moved across the lower display 114, the application maximization function is completed. At this point, data from the first application 1.1 is displayed on both displays 110, 114 in either an LD output configuration or an Lmax output configuration 628. Maximization rules may be referenced by the window management module 532 to determine how the application data should be presented when maximized. Such maximization rules may be maintained within the window stack 550 or be made available to the window management module 532 via a data lookup function. In the example depicted in Fig. 7A, the first application 1.1 is depicted in a an LD output configuration such that a first set of data from the first application 1.1 is presented via display 110 in a first landscape output configuration 612 while a second set of data (not inclusive of the first set of data) from the first application 1.1 is presented via display 114 in a second landscape output configuration 616. If an Lmax output configuration 628 were utilized, then at least some data from the first application 1.1 would be presented simultaneously by both displays 110, 114.

[0011] A second gesture sequence 712 is depicted in Fig. 7B. Like the first gesture sequence 700, the second gesture sequence 712 may be implemented on any component of the screens 104, 108. The second gesture sequence 712 begins similar to the first gesture sequence 700 except that data from a minimized application is initially displayed on a lower display (e.g., display 114) rather than the upper display (e.g., display 110). As can be appreciated, the actual display on which the minimized application is presented may depend upon the orientation of the device 100 (e.g., whether display 110 is above display 114 or vice versa). Moreover the data from the application may be displayed on the lower display according to minimization display rules determined by one or both of the MDM module 524 and window management module 532 for the application.

[0012] The concepts of desktop stacks may still be applied and, in some embodiments, the window management module 532 may determine that since data from the application is being displayed on the lower display in a second landscape output configuration 616, a first desktop

display 114 while data from the first application 1.1 is presented in a minimized state on the other display 110.

[0004] The first gesture sequence 700 continues when a first user input 704 of a predetermined type is detected by the device 100. In some embodiments, the first user input 704 may correspond to a drag gesture. The drag gesture may be detected in a touch sensitive display 110, 114, a gesture capture region 120, 124, a configurable area 112, 116, or any other part of one or both screens 104, 108. In some embodiments, the drag gesture is detected in a gesture capture region 120, 124 of either the primary screen 104 or the secondary screen 108. Specifically, the drag gesture may be moving in a direction that is generally away from the top edge of the upper display 110 and toward the bottom edge of the lower display 114 (e.g., a downward drag gesture). In a more specific embodiment, the drag gesture may be detected on the primary screen 104 (e.g., the screen on which the data from the currently minimized application is being depicted).

[0005] Although a single gesture is described as being the first user gesture 704, those of ordinary skill in the art will appreciate that the first gesture 704 may actually correspond to any single gesture or combination of gestures described herein or otherwise known to those of ordinary skill in the art.

[0006] The detections of the first gesture 704 is detected and interpreted by the task management module 540. The gesture interpretation made by the task management module 540 is then forwarded to the MDM module 524 which correlates the received first gesture 704 to one or more predetermined display outputs or display actions.

[0007] Specifically, once the MDM module 524 registers the drag gesture, the MDM module 524 may correlate the detected gesture to a particular display control action. Even more specifically, the MDM module 524 may determine that the downward drag gesture corresponds to an instruction to maximize the display of the first application 1.1 currently being displayed on display 110. Accordingly, the MDM module 524 may generate one or more instructions for creating a transition area 708 that contains data from the first application 1.1 being maximized.

[0008] The instructions generated by the MDM module 524 may be provided back to the task management module 540 which can then provide the instructions to the window management module 532. The window management module 524 may then invoke the changes to the displays 110, 114 according to the instructions generated by the MDM module 524.

display D1 may be presented on the other display 110. As can be appreciated, if the concept of a desktop stack were followed, then a second desktop display D2 may be presented on the lower display 114 if the desktop were completely presented by the device 100. Rather than depicting the first desktop display D1, the display 110 may be depicting data from an application other than the application being displayed on display 114.

[0013] The second gesture sequence 712 continues when a first user input 716 of a predetermined type is detected by the device 100 and registered at the task management module 540. The first user input 716 may be similar to the first user input 704 except that is may have a different direction of motion (e.g., upward rather than downward). In some embodiments, the first user input 716 may correspond to a drag gesture. The drag gesture may be detected in a touch sensitive display 110, 114, a gesture capture region 120, 124, a configurable area 112, 116, or any other part of one or both screens 104, 108. In some embodiments, the drag gesture is detected in a gesture capture region 120, 124 of either the primary screen 104 or the secondary screen 108. Specifically, the drag gesture may be moving in a direction that is generally away from the bottom edge of the lower display 114 and toward the top edge of the upper display 110 (e.g., an upward drag gesture). In a more specific embodiment, the drag gesture may be detected on the secondary screen 108 (e.g., the screen on which the data from the currently minimized application is being depicted).

[0014] Although a single gesture is described as being the first user gesture 716, those of ordinary skill in the art will appreciate that the first gesture 716 may actually correspond to any single gesture or combination of gestures described herein or otherwise known to those of ordinary skill in the art.

[0015] Once the task management module 540 has interpreted the first user gesture 716 and provided such information to the MDM module 524, the MDM module 524 detects and registers the drag gesture. Thereafter, the MDM module 524 may correlate the detected gesture to a particular display control action. Specifically, the MDM module 524 may determine that the upward drag gesture corresponds to an instruction to maximize the display of the first application 1.1 currently being displayed on display 114. Accordingly, the MDM module 524 may generate an instruction to create a transition area 708 that contains data from the first application 1.1 being maximized. This instruction may then be provided to the window management module 532 via the task management module 540.

[0016] The transition area 708 may progressively grow across the upper display 110 until data from the first application 1.1 being maximized fills both displays 110, 114. Until the transition area 708 has completely filled the upper display 110, the data previously presented on the upper display 110 (e.g., the first desktop display D1) may be presented in the remaining portion of the display 110 not occupied by the transition area 708.

[0017] In some embodiments, after the transition area 708 has completely moved across the upper display 110, the application maximization function is completed. At this point, data from the first application 1.1 is displayed on both displays 110, 114 in either an LD output configuration or an Lmax output configuration 628. Maximization rules may be referenced by the window management module 532 to determine how the application data should be presented when maximized. Such maximization rules may be maintained within the window management module 532 or be made available to the window management module 532 via a data lookup function. In the example depicted in Fig. 7B, the first application 1.1 is depicted in a an LD output configuration such that a first set of data from the first application 1.1 is presented via display 110 in a first landscape output configuration 612 while a second set of data (not inclusive of the first set of data) from the first application 1.1 is presented via display 114 in a second landscape output configuration 616. If an Lmax output configuration 628 were utilized, then at least some data from the first application 1.1 would be presented simultaneously by both displays 110, 114.

[0018] A third gesture sequence 720 is depicted in Fig. 7C. Like the previous gesture sequences 700, 712, the third gesture sequence 720 may be implemented on any component of the screens 104, 108. The third gesture sequence 720 begins similarly to the second gesture sequence 712 in that data from a first application 1.1 is presented in a minimized state on one of the displays 114 whereas data from another application or perhaps the desktop is presented on the other display 110. Although data from the first application 1.1 is depicted as being initially displayed on the lower display (e.g., display 114), it should be appreciated that the third gesture sequence 720 may also be implemented when data from the first application 1.1 is initially presented on the upper display (e.g., display 110) and other data is presented on the other display. In either scenario, data from the first application 1.1 may be displayed on the display 110 or 114 according to minimization display rules determined by the window management module 532 for the application.

[0019] The third gesture sequence 720 continues when a first user input 724 of a predetermined type is detected by the device 100. The first user input 724 may be similar to the other user inputs 704, 716, except that the first user input 724 may correspond to an instruction to move the data from the first application 1.1, rather than maximizing data from the first application 1.1. As a non-limiting example, the first user input 724 may correspond to a flicking gesture in either an upward or downward direction, depending upon the location of the first application 1.1. The flicking gesture may be captured by any part of a screen 104 or 108.

[0020] Although a single gesture is described as being the first user gesture 724, those of ordinary skill in the art will appreciate that the first gesture 724 may actually correspond to any single gesture or combination of gestures described herein or otherwise known to those of ordinary skill in the art.

[0021] The first gesture 724 is registered as the task management module 540 and interpreted. Thereafter, the task management module 540 provides the MDM module 540 with information about the received first gesture 724. Once the MDM module 524 detects and registers the flicking gesture, the MDM module 524 may correlate the detected gesture to a particular display control action. Specifically, the MDM module 524 may determine that the flicking gesture corresponds to an instruction to move the display of the first application 1.1 from one display to the other display (e.g., from display 114 to display 110). Accordingly, the MDM module 524 may generate an instruction to create a transition area 728 of a fixed dimension that contains data from the first application 1.1 being moved.

[0022] The instruction(s) generated by the MDM module 524 may be provided to the window management module 532 which references display rules and executes the display changes. Specifically, the window management module 532 may cause the transition area 728 to migrate away from one display (e.g., display 114) toward the other display (e.g., display 110) until data from the first application 1.1 being moved has been moved to the other display (e.g., display 110). While the transition area 728 is moving between displays, data previously presented on the upper display 110 (e.g., the first desktop display D1) may be presented in the remaining portion of the display 110 not occupied by the transition area 728. Similarly, data presented on the lower display 114 (e.g., the second desktop display D2) may be presented in the growing portion of the display 114.

[0023] Afrer the first application 1.1 has moved from one display to another display, the sequence may be considered completed. At this time data from the first application 1.1 may now be presented on a different display 110 in a first landscape output configuration 612 and a different desktop display (e.g., the second desktop display D2) may be presented on display 114 in a second landscape output configuration 616.

[0024] A fourth gesture sequence 732 is depicted in Fig. 7D. The fourth gesture sequence 732 corresponds invokes a display action that minimizes an application rather than maximizing or moving an application. The fourth gesture sequence 732, like other gesture sequences described herein, can be implemented on any component of the screens 104, 108 without departing from the scope of the present disclosure.

[0025] The fourth gesture sequence 732 begins with a first application 1.1 being displayed in a maximized state on both displays 110, 114. In some embodiments, maximization rules may be referenced to determine how the first application 1.1 is displayed across multiple screens 104, 108. The first application 1.1 may be originally displayed in either an LD output configuration or an Lmax output configuration 628.

[0026] The fourth gesture sequence 744 continues when a first user input 736 of a predetermined type is detected by the device 100. The first user input 736 may be similar to other user inputs described herein. In some embodiments, the first user input 736 may correspond to either a drag or a flick gesture. The drag or flick gesture may be detected in a touch sensitive display 110, 114, a gesture capture region 120, 124, a configurable area 112, 116, or any other part of one or both screens 104, 108. In some embodiments, the drag or flick gesture is detected in a gesture capture region 120, 124 of either the primary screen 104 or the secondary screen 108. Specifically, a first drag or flick gesture 736 is depicted in Fig. 7D that invokes one type of minimization display action where the first application 1.1 is minimized in a downward direction. A second drag or flick gesture 744 is depicted in Fig. 7E in connection with a fifth gesture sequence 740 that invokes another type of minimization display action where the first application 1.1 is minimized in an upward direction. Accordingly, the minimization display actions of Figs. 7D and 7E may be similar except that one display action moves the first application 1.1 to one display 110, whereas the other display action moves the first application 1.1 to another display 114.

[0027] Although a single gesture is described as being the first user gesture 736 or 744, those of ordinary skill in the art will appreciate that the first gesture 736, 744 may actually correspond to any single gesture or combination of gestures described herein or otherwise known to those of ordinary skill in the art.

[0028] In either minimization display action, once the MDM module 524 detects and registers the drag or flicking gesture 736, 744 (as received from the task management module 540), the MDM module 524 may correlate the detected gesture to a particular display control action. Specifically, the MDM module 524 may determine that the drag or flicking gesture corresponds to an instruction to minimize the display of the first application 1.1 to a single display (or sub-component of a single display). Accordingly, the MDM module 524 may generate one or more instructions to create a transition area 728 of a fixed dimension that contains data from the first application 1.1 being moved.

[0029] The instructions generated by the MDM module 524 may be provided to the window management module 532 which actually renders and causes the transition area 728 to migrate away from one display toward the other display until data from the first application 1.1 being moved has been moved to the other display. The direction in which the transition area 728 moves may be the same as the direction of the drag or flicking gesture. While the transition area 728 is moving between displays, desktop displays may be partially rendered on both the displays 110, 114. Specifically, some of a first desktop display D1 may be rendered on one display 110 while some of a second desktop display D2 may be rendered on another display 114.

[0030] Afrer the first application 1.1 has been minimized and reformatted for display on a single screen 104 or 108, the minimization display action may be considered completed. At this time data from the first application 1.1 may now be presented on a single display 110 or 114 in a landscape output configuration 612 or 616 and a desktop display may be presented on the other display 110 or 114. The manner in which the first application 1.1 and desktop is displayed may depend upon minimization rules that are available to and referenced by the window management module 532.

[0031] Referring now to Fig. 8, a first display control method 800 will be described in accordance with at least some embodiments of the present disclosure. The method 800 is initiated at step 804. Specifically, a device 100 may be initially in an open landscape state 348.

The method continues when a user input (e.g., a gesture or combination of gestures) is received at the device 100 and registered by the MDM module 524 (step 808).

[0032] Thereafter, the MDM module 524 may analyze the received information about the user input and determine that the user input corresponds to application-maximization instructions (step 812). The MDM module 524 may further determine what specific type of application-maximization instructions have been received (e.g., direction of drag or flick, location of received instructions, etc.) and may correlate the specific type of user input to maximization rules for the application being maximized (step 816). Accordingly, based on the maximization rules for the application, the MDM module 524 will instruct the window management module 532 to reformat the display of the application to at least partially occupy both displays 110, 114, unless the maximization instructions for the application dictate otherwise. Examples of such display control actions are depicted in Figs. 7A and 7B.

[0033] Referring now to Fig. 9, a second display control method 900 will be described in accordance with at least some embodiments of the present disclosure. The method 900 is initiated at step 904. Specifically, a device 100 may be initially in an open landscape state 348. The method continues when a user input (e.g., a gesture or combination of gestures) is received at the device 100 and registered by the MDM module 524 (step 908).

[0034] Thereafter, the MDM module 524 may analyze the received information about the user input and determine that the user input corresponds to application-minimization instructions (step 912). The MDM module 524 may further determine what specific type of application-maximization instructions have been received (e.g., direction of drag or flick, location of received instructions, etc.) and may correlate the specific type of user input to minimization rules for the application being minimized (step 916). Accordingly, based on the minimization rules for the application, the MDM module 524 will instruct the window management module 532 to reformat the display of the application to occupy only a single display 110 or 114 (or a portion thereof), unless the minimization instructions for the application dictate otherwise. Examples of such display control actions are depicted in Figs. 7D and 7E.

[0035] The exemplary systems and methods of this disclosure have been described in relation to display control mechanisms. However, to avoid unnecessarily obscuring the present disclosure, the preceding description omits a number of known structures and devices. This omission is not to be construed as a limitation of the scopes of the claims. Specific details are set

forth to provide an understanding of the present disclosure. It should however be appreciated that the present disclosure may be practiced in a variety of ways beyond the specific detail set forth herein.

[0036] Furthermore, while the exemplary aspects, embodiments, and/or configurations illustrated herein show the various components of the system collocated, certain components of the system can be located remotely, at distant portions of a distributed network, such as a LAN and/or the Internet, or within a dedicated system. Thus, it should be appreciated, that the components of the system can be combined in to one or more devices, such as a computer, laptop, netbook, smart phone, mobile device, etc., or collocated on a particular node of a distributed network, such as an analog and/or digital telecommunications network, a packet-switch network, or a circuit-switched network. It will be appreciated from the preceding description, and for reasons of computational efficiency, that the components of the system can be arranged at any location within a distributed network of components without affecting the operation of the system. For example, the various components can be located in a switch such as a PBX and media server, gateway, in one or more communications devices, at one or more users' premises, or some combination thereof. Similarly, one or more functional portions of the system could be distributed between a telecommunications device(s) and an associated computing device.

[0037] Furthermore, it should be appreciated that the various links connecting the elements can be wired or wireless links, or any combination thereof, or any other known or later developed element(s) that is capable of supplying and/or communicating data to and from the connected elements. These wired or wireless links can also be secure links and may be capable of communicating encrypted information. Transmission media used as links, for example, can be any suitable carrier for electrical signals, including coaxial cables, copper wire and fiber optics, and may take the form of acoustic or light waves, such as those generated during radio-wave and infra-red data communications.

[0038] Also, while the flowcharts have been discussed and illustrated in relation to a particular sequence of events, it should be appreciated that changes, additions, and omissions to this sequence can occur without materially affecting the operation of the disclosed embodiments, configuration, and aspects.

[0039] A number of variations and modifications of the disclosure can be used. It would be possible to provide for some features of the disclosure without providing others.

[0040] In yet another embodiment, the systems and methods of this disclosure can be implemented in conjunction with a special purpose computer, a programmed microprocessor or microcontroller and peripheral integrated circuit element(s), an ASIC or other integrated circuit, a digital signal processor, a hard-wired electronic or logic circuit such as discrete element circuit, a programmable logic device or gate array such as PLD, PLA, FPGA, PAL, special purpose computer, any comparable means, or the like. In general, any device(s) or means capable of implementing the methodology illustrated herein can be used to implement the various aspects of this disclosure. Exemplary hardware that can be used for the disclosed embodiments, configurations and aspects includes computers, handheld devices, telephones (e.g., cellular, Internet enabled, digital, analog, hybrids, and others), and other hardware known in the art. Some of these devices include processors (e.g., a single or multiple microprocessors), memory, nonvolatile storage, input devices, and output devices. Furthermore, alternative software implementations including, but not limited to, distributed processing or component/object distributed processing, parallel processing, or virtual machine processing can also be constructed to implement the methods described herein.

[0041] In yet another embodiment, the disclosed methods may be readily implemented in conjunction with software using object or object-oriented software development environments that provide portable source code that can be used on a variety of computer or workstation platforms. Alternatively, the disclosed system may be implemented partially or fully in hardware using standard logic circuits or VLSI design. Whether software or hardware is used to implement the systems in accordance with this disclosure is dependent on the speed and/or efficiency requirements of the system, the particular function, and the particular software or hardware systems or microprocessor or microcomputer systems being utilized.

[0042] In yet another embodiment, the disclosed methods may be partially implemented in software that can be stored on a storage medium, executed on programmed general-purpose computer with the cooperation of a controller and memory, a special purpose computer, a microprocessor, or the like. In these instances, the systems and methods of this disclosure can be implemented as program embedded on personal computer such as an applet, JAVA® or CGI script, as a resource residing on a server or computer workstation, as a routine embedded in a

dedicated measurement system, system component, or the like. The system can also be implemented by physically incorporating the system and/or method into a software and/or hardware system.

[0043] Although the present disclosure describes components and functions implemented in the aspects, embodiments, and/or configurations with reference to particular standards and protocols, the aspects, embodiments, and/or configurations are not limited to such standards and protocols. Other similar standards and protocols not mentioned herein are in existence and are considered to be included in the present disclosure. Moreover, the standards and protocols mentioned herein and other similar standards and protocols not mentioned herein are periodically superseded by faster or more effective equivalents having essentially the same functions. Such replacement standards and protocols having the same functions are considered equivalents included in the present disclosure.

[0044] The present disclosure, in various aspects, embodiments, and/or configurations, includes components, methods, processes, systems and/or apparatus substantially as depicted and described herein, including various aspects, embodiments, configurations embodiments, subcombinations, and/or subsets thereof. Those of skill in the art will understand how to make and use the disclosed aspects, embodiments, and/or configurations afrer understanding the present disclosure. The present disclosure, in various aspects, embodiments, and/or configurations, includes providing devices and processes in the absence of items not depicted and/or described herein or in various aspects, embodiments, and/or configurations hereof, including in the absence of such items as may have been used in previous devices or processes, e.g., for improving performance, achieving ease and\or reducing cost of implementation.

[0045] The foregoing discussion has been presented for purposes of illustration and description. The foregoing is not intended to limit the disclosure to the form or forms disclosed herein. In the foregoing Detailed Description for example, various features of the disclosure are grouped together in one or more aspects, embodiments, and/or configurations for the purpose of streamlining the disclosure. The features of the aspects, embodiments, and/or configurations of the disclosure may be combined in alternate aspects, embodiments, and/or configurations other than those discussed above. This method of disclosure is not to be interpreted as reflecting an intention that the claims require more features than are expressly recited in each claim. Rather, as the following claims reflect, inventive aspects lie in less than all features of a single foregoing

disclosed aspect, embodiment, and/or configuration. Thus, the following claims are hereby incorporated into this Detailed Description, with each claim standing on its own as a separate preferred embodiment of the disclosure.

[0046] Moreover, though the description has included description of one or more aspects, embodiments, and/or configurations and certain variations and modifications, other variations, combinations, and modifications are within the scope of the disclosure, e.g., as may be within the skill and knowledge of those in the art, after understanding the present disclosure. It is intended to obtain rights which include alternative aspects, embodiments, and/or configurations to the extent permitted, including alternate, interchangeable and/or equivalent structures, functions, ranges or steps to those claimed, whether or not such alternate, interchangeable and/or equivalent structures, functions, ranges or steps are disclosed herein, and without intending to publicly dedicate any patentable subject matter.

What is claimed is:

1. A method of controlling data displayed by a multi-screen device, comprising:

receiving a first user input that corresponds to instructions to maximize a first application displayed on a first screen of the dual-screen device;

referencing maximization rules for the first application; and

based on the maximization rules for the first application, causing data from the first application to be simultaneously displayed on both the first screen and a second screen of the dual-screen device.

2. The method of claim 1, wherein the maximization rules for the first application define that the first application is to be displayed on all of a display of the first screen as well as all of a display of the second screen when maximized and when the dual-screen device is in an open landscape state.

3. The method of claim 2, wherein the first application is maximized in a dual-landscape output configuration such that a first set of data from the first application is displayed on the display of the first screen and a second set of data from the first application is displayed on the display of the second screen.

4. The method of claim 2, wherein the first application is maximized in a landscape max output configuration such that a first set of data from the first application is displayed simultaneously on both the display of the first screen and the display of the second screen.

5. The method of claim 4, the method further comprising:

generating a transition area that progressively fills the display of the second screen with data from the first application; and

causing the transition area to replace a desktop display that was previously presented on the display of the second screen.

6. The method of claim 5, wherein the desktop display that was previously presented on the display of the second screen corresponds to one of a plurality of desktop displays in a desktop display stack and wherein another desktop display in the desktop display stack is assigned to the first screen and is covered by the first application.

7. The method of claim 1, further comprising:

receiving a second user input that corresponds to instructions to minimize the first application displayed on the first and second screen of the dual-screen device; and

reformatting the first application such that it is displayed on the first screen similar to the way in which the first application was displayed on the first screen prior to receiving the first user input.

8. The method of claim 1, wherein the first user input corresponds to at least one of a flick and drag gesture.

9. The method of claim 1, wherein the first user input is received in a gesture capture region of the first screen and wherein the gesture capture region of the first screen is separate from a display area of the first screen.

10. A non-transitory computer-readable medium having stored thereon instructions that cause a computing system to execute a method, the instructions comprising:

instructions configured to analyze a first user input received at a dual-screen device and determine that the first user input corresponds to instructions to maximize a first application displayed on a first screen of the dual-screen device;

instructions configuration to reference maximization rules for the first application; and

instructions configured to invoke the maximization of the first application such that data from the first application is simultaneously displayed on both the first screen and a second screen of the dual-screen device.

11. The computer-readable medium of claim 10, wherein the first screen corresponds to a primary screen and wherein the second screen corresponds to a secondary screen.

12. The computer-readable medium of claim 10, wherein the maximization rules for the first application define that the first application is to be displayed on all of a display of the first screen as well as all of a display of the second screen when maximized and when the dual-screen device is in an open landscape state.

13. The computer-readable medium of claim 12, wherein the first application is maximized in a dual-landscape output configuration such that a first set of data from the first application is displayed on the display of the first screen and a second set of data from the first application is displayed on the display of the second screen.

14. The computer-readable medium of claim 12, wherein the first application is maximized in a landscape max output configuration such that a first set of data from the first application is displayed simultaneously on both the display of the first screen and the display of the second screen.

15. The computer-readable medium of claim 14, further comprising:

instructions configured to generate a transition area that progressively fills the display of the second screen with data from the first application; and

instructions configured to cause the transition area to replace a desktop display that was previously presented on the display of the second screen.

16. A dual-screen user device, comprising:

a first screen including a first display area;

a second screen including a second display area; and

a computer-readable medium having instructions stored thereon that include:

a first set of instructions configured to analyze a first user input received at a dual-screen device and determine that the first user input corresponds to instructions to maximize a first application displayed on the display of the first screen;

a second set of instructions configuration to reference maximization rules for the first application; and

a third set of instructions configured to invoke the maximization of the first application such that data from the first application is simultaneously displayed on the both the first display area and the second display area.

17. The device of claim 16, wherein the first screen comprises a gesture capture region that is separate from the first display area of the first screen and wherein the first user input comprises at least one of a drag and flick gesture received in the gesture capture region that is directed toward the second screen.

18. The device of claim 17, wherein the first user input comprises a drag gesture and wherein the first application replaces a second desktop display that was displayed on the second display area prior to receiving the first user input.

19. The device of claim 17, wherein the data from the first application is automatically displayed on both the first and second display areas in a landscape max output configuration.

20. The device of claim 17, wherein the data from the first application is automatically displayed on both the first and second display areas in a landscape dual output configuration.

ABSTRACT

A dual-screen user device and methods for controlling data displayed thereby are disclosed. Specifically, the data displayed by the multiple screens of the dual-screen user device is conditioned upon the type of user gesture or combination of user gestures detected. The display controls described herein can correlate user inputs received in a gesture capture region to one or more display actions, which may include maximization, minimization, or reformatting instructions.

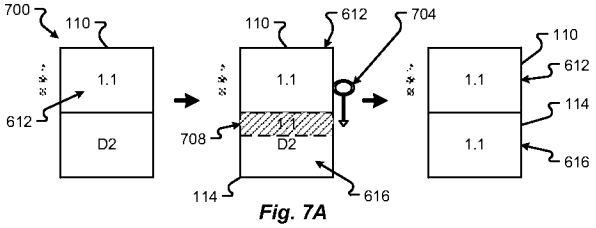


Fig. 7A

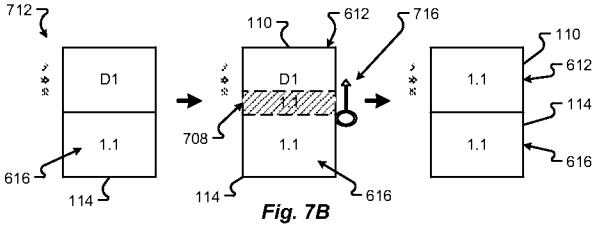


Fig. 7B

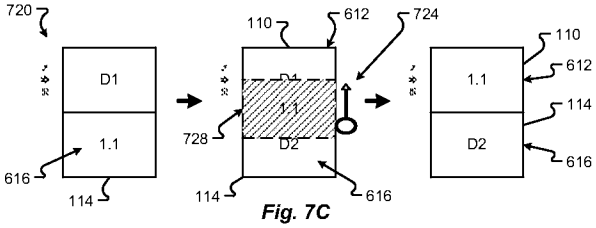


Fig. 7C

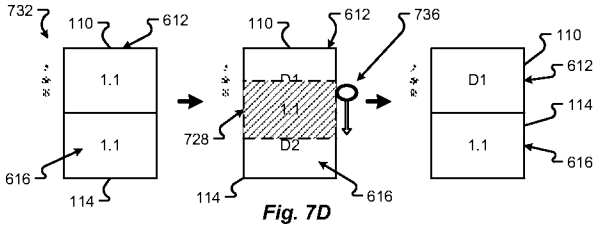


Fig. 7D

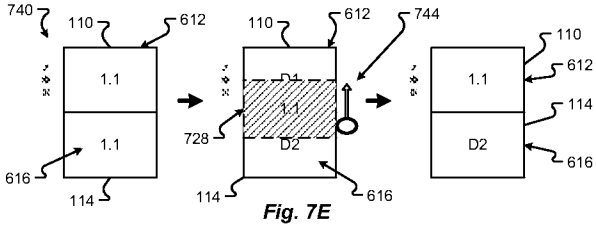


Fig. 7E

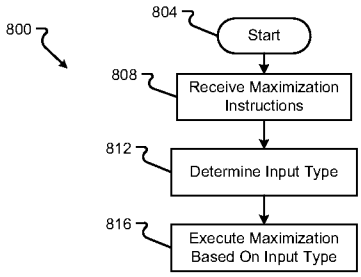


Fig. 8

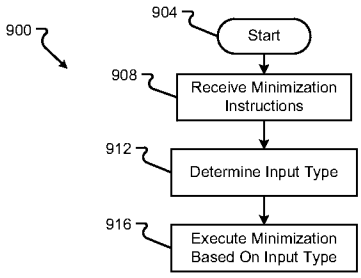


Fig. 9

MANAGING EXPOSE VIEWS IN DUAL DISPLAY COMMUNICATION DEVICES

[0100] The device 100 manages desktops and/or windows with at least one window stack 1700, 1728, as shown in Figs. 9A and 9B. A window stack 1700, 1728 is a logical arrangement of active and/or inactive windows for a multi-screen device. For example, the window stack 1700, 1728 may be logically similar to a deck of cards, where one or more windows or desktops are arranged in order, as shown in Figs. 9A and 9B. An active window is a window that is currently being displayed on at least one of the touch sensitive displays 110, 114. For example, windows 104 and 108 are active windows and are displayed on touch sensitive displays 110 and 114. An inactive window is a window that was opened and displayed but is now "behind" an active window and not being displayed. In embodiments, an inactive window may be for an application that is suspended, and thus, the window is not displaying active content. For example, windows 1712, 1716, 1720, and 1724 are inactive windows.

[0101] A window stack 1700, 1728 may have various arrangements or organizational structures. In the embodiment shown in Fig. 9A, the device 100 includes a first stack 1760 associated with a first touch sensitive display 110 and a second stack associated with a second touch sensitive display 114. Thus, each touch sensitive display 110, 114 can have an associated window stack 1760, 1764. These two window stacks 1760, 1764 may have different numbers of windows arranged in the respective stacks 1760, 1764. Further, the two window stacks 1760, 1764 can also be identified differently and managed separately. Thus, the first window stack 1760 can be arranged in order from a first window 1704 to a next window 1720 to a last window 1724 and finally to a desktop 1722, which, in embodiments, is at the "bottom" of the window stack 1760. In embodiments, the desktop 1722 is not always at the "bottom" as application windows can be arranged in the window stack below the desktop 1722, and the desktop 1722 can be brought to the "top" of a stack over other windows during a desktop reveal. Likewise, the second stack 1764 can be arranged from a first window 1708 to a next window 1712 to a last window 1716, and finally to a desktop 1718, which, in embodiments, is a single desktop area, with desktop 1722, under all the windows in both window stack 1760 and window stack 1764. A logical data structure for managing the two window stacks 1760, 1764 may be as described in conjunction with Fig. 10.

[0102] Another arrangement for a window stack 1728 is shown in Fig. 9B. In this embodiment, there is a single window stack 1728 for both touch sensitive displays 110, 114.

Thus, the window stack 1728 is arranged from a desktop 1758 to a first window 1744 to a last window 1756. A window can be arranged in a position among all windows without an association to a specific touch sensitive display 110, 114. In this embodiment, a window is in the order of windows. Further, at least one window is identified as being active. For example, a single window may be rendered in two portions 1732 and 1736 that are displayed on the first touch sensitive screen 110 and the second touch sensitive screen 114. The single window may only occupy a single position in the window stack 1728 although it is displayed on both displays 110, 114.

[0103] Yet another arrangement of a window stack 1760 is shown in Figs. 9C through 9E. The window stack 1760 is shown in three "elevation" views. In Fig. 9C, the top of the window stack 1760 is shown. Two sides of the window stack 1760 are shown in Figs. 9D and 9E. In this embodiment, the window stack 1760 resembles a stack of bricks. The windows are stacked on each other. Looking from the top of the window stack 1760 in Fig, 9C, only the top most windows in the window stack 1760 are seen in different portions of the composite display 1764. The composite display 1764 represents a logical model for the entire display area of the device 100, which can include touch sensitive display 110 and touch sensitive display 114. A desktop 1786 or a window can occupy part or all of the composite display 1764.

[0104] In the embodiment shown, the desktop 1786 is the lowest display or "brick" in the window stack 1760. Thereupon, window 1 1782, window 2 1782, window 3 1768, and window 4 1770 are layered. Window 1 1782, window 3 1768, window 2 1782, and window 4 1770 only occupy a portion of the composite display 1764. Thus, another part of the stack 1760 includes window 8 1774 and windows 5 through 7 shown in section 1790. Only the top window in any portion of the composite display 1764 is actually rendered and displayed. Thus, as shown in the top view in Fig. 9C, window 4 1770, window 8 1774, and window 3 1768 are displayed as being at the top of the display in different portions of the window stack 1760. A window can be dimensioned to occupy only a portion of the composite display 1760 to "reveal" windows lower in the window stack 1760. For example, window 3 1768 is lower in the stack than both window 4 1770 and window 8 1774 but is still displayed. A logical data structure to manage the window stack can be as described in conjunction with Fig. 10.

[0105] When a new window is opened, the newly activated window is generally positioned at the top of the stack. However, where and how the window is positioned within the stack can be

a function of the orientation of the device 100, the context of what programs, functions, software, etc. are being executed on the device 100, how the stack is positioned when the new window is opened, etc. To insert the window in the stack, the position in the stack for the window is determined and the touch sensitive display 110, 114 to which the window is associated may also be determined. With this information, a logical data structure for the window can be created and stored. When user interface or other events or tasks change the arrangement of windows, the window stack(s) can be changed to reflect the change in arrangement. It should be noted that these same concepts described above can be used to manage the one or more desktops for the device 100.

[0106] A logical data structure 1800 for managing the arrangement of windows or desktops in a window stack is shown in Fig. 10. The logical data structure 1800 can be any data structure used to store data whether an object, record, file, etc. The logical data structure 1800 can be stored in any type of database or data storage system, regardless of protocol or standard. In embodiments, the logical data structure 1800 includes one or more portions, fields, attributes, etc. that store data in a logical arrangement that allows for easy storage and retrieval of the information. Hereinafter, these one or more portions, fields, attributes, etc. shall be described simply as fields. The fields can store data for a window identifier 1804, dimensions 1808, a stack position identifier 1812, a display identifier 1816, and/or an active indicator 1820. Each window in a window stack can have an associated logical data structure 1800. While only a single logical data structure 1800 is shown in Fig. 10, there may be more or fewer logical data structures 1800 used with a window stack (based on the number of windows or desktops in the stack), as represented by ellipses 1824. Further, there may be more or fewer fields than those shown in Fig. 10, as represented by ellipses 1828.

[0107] A window identifier 1804 can include any identifier (ID) that uniquely identifies the associated window in relation to other windows in the window stack. The window identifier 1804 can be a globally unique identifier (GUID), a numeric ID, an alphanumeric ID, or other type of identifier. In embodiments, the window identifier 1804 can be one, two, or any number of digits based on the number of windows that can be opened. In alternative embodiments, the size of the window identifier 1804 may change based on the number of windows opened. While the window is open, the window identifier 1804 may be static and remain unchanged.

[0108] Dimensions 1808 can include dimensions for a window in the composite display 1760. For example, the dimensions 1808 can include coordinates for two or more corners of the window or may include one coordinate and dimensions for the width and height of the window. These dimensions 1808 can delineate what portion of the composite display 1760 the window may occupy, which may the entire composite display 1760 or only part of composite display 1760. For example, window 4 1770 may have dimensions 1880 that indicate that the window 1770 will occupy only part of the display area for composite display 1760, as shown in Figs. 9C through 9E. As windows are moved or inserted in the window stack, the dimensions 1808 may change.

[0109] A stack position identifier 1812 can be any identifier that can identify the position in the stack for the window or may be inferred from the window's control record within a data structure, such as a list or a stack. The stack position identifier 1812 can be a GUID, a numeric ID, an alphanumeric ID, or other type of identifier. Each window or desktop can include a stack position identifier 1812. For example, as shown in Fig. 9A, window 1 1704 in stack 1 1760 can have a stack position identifier 1812 of 1 identifying that window 1704 is the first window in the stack 1760 and the active window. Similarly, window 6 1724 can have a stack position identifier 1812 of 3 representing that window 1724 is the third window in the stack 1760. Window 2 1708 can also have a stack position identifier 1812 of 1 representing that window 1708 is the first window in the second stack 1764. As shown in Fig. 9B, window 1 1744 can have a stack position identifier 1812 of 1, window 3, rendered in portions 1732 and 1736, can have a stack position identifier 1812 of 3, and window 6 1756 can have a stack position identifier 1812 of 6. Thus, depending on the type of stack, the stack position identifier 1812 can represent a window's location in the stack.

[0110] A display identifier 1816 can identify that the window or desktop is associated with a particular display, such as the first display 110 or the second display 114, or the composite display 1760 composed of both displays. While this display identifier 1816 may not be needed for a multi-stack system, as shown in Fig. 9A, the display identifier 1816 can indicate whether a window in the serial stack of Fig. 9B is displayed on a particular display. Thus, window 3 may have two portions 1732 and 1736 in Fig. 9B. The first portion 1732 may have a display identifier 1816 for the first display while the second portion 1736 may have a display identifier 1816 for the second display 114. However, in alternative embodiments, the window may have

two display identifier 1816 that represent that the window is displayed on both of the displays 110, 114, or a display identifier 1816 identifying the composite display. In another alternate embodiment, the window may have a single display identifier 1816 to represent that the window is displayed on both of the displays 110, 114.

[0111] Similar to the display identifier 1816, an active indicator 1820 may not be needed with the dual stack system of Fig. 9A, as the window in stack position 1 is active and displayed. In the system of Fig. 9B, the active indicator 1820 can indicate which window(s) in the stack is being displayed. Thus, window 3 may have two portions 1732 and 1736 in Fig. 9. The first portion 1732 may have an active indicator 1820 while the second portion 1736 may also have an active indicator 1820. However, in alternative embodiments, window 3 may have a single active indicator 1820. The active indicator 1820 can be a simple flag or bit that represents that the window is active or displayed.

[0112] An embodiment of a method 900 for creating a window stack is shown in Fig. 11. While a general order for the steps of the method 1900 is shown in Fig. 11. Generally, the method 1900 starts with a start operation 1904 and ends with an end operation 1928. The method 1900 can include more or fewer steps or can arrange the order of the steps differently than those shown in Fig. 11. The method 1900 can be executed as a set of computer-executable instructions executed by a computer system and encoded or stored on a computer readable medium. Hereinafrer, the method 1900 shall be explained with reference to the systems, components, modules, software, data structures, user interfaces, etc. described in conjunction with Figs. 1-10.

[0113] A multi-screen device 100 can receive activation of a window, in step 1908. In embodiments, the multi-screen device 100 can receive activation of a window by receiving an input from the touch sensitive display 110 or 114, the configurable area 112 or 116, a gesture capture region 120 or 124, or some other hardware sensor operable to receive user interface inputs. The processor may execute the Task Management Module 540 may receive the input. The Task Management Module 540 can interpret the input as requesting an application task to be executed that will open a window in the window stack.

[0114] In embodiments, the Task Management Module 540 places the user interface interaction in the task stack 552 to be acted upon by the Display Configuration Module 568 of the Multi-Display Management Module 524. Further, the Task Management Module 540 waits

for information from the Multi-Display Management Module 524 to send instructions to the Window Management Module 532 to create the window in the window stack.

[0115] The Multi-Display Management Module 524, upon receiving instruction from the Task Management Module 540, determines to which touch portion of the composite display 1760, the newly activated window should be associated, in step 1912. For example, window 4 1770 is associated with the a portion of the composite display 1764 In embodiments, the device state module 574 of the Multi-Display Management Module 524 may determine how the device is oriented or in what state the device is in, e.g., open, closed, portrait, etc. Further, the preferences module 572 and/or requirements module 580 may determine how the window is to be displayed. The gesture module 576 may determine the user's intentions about how the window is to be opened based on the type of gesture and the location of where the gesture is made.

[0116] The Display Configuration Module 568 may use the input from these modules and evaluate the current window stack 1760 to determine the best place and the best dimensions, based on a visibility algorithm, to open the window. Thus, the Display Configuration Module 568 determines the best place to put the window at the top of the window stack 1760, in step 1916. The visibility algorithm, in embodiments, determines for all portions of the composite display, which windows are at the top of the stack. For example, the visibility algorithm determines that window 3 1768, window 4 1770, and window 8 1774 are at the top of the stack 1760 as viewed in Figs. 9C through 9E. Upon determining where to open the window, the Display Configuration Module 568 can assign a display identifier 816 and possibly dimensions 808 to the window. The display identifier 816 and dimensions 808 can then be sent back to the Task Management Module 540. The Task Management Module 540 may then assign the window a stack position identifier 812 indicating the windows position at the top of the window stack.

[0117] In embodiments, the Task Management Module 540 sends the window stack information and instructions to render the window to the Window Management Module 532. The Window Management Module 532 and the Task Management Module 540 can create the logical data structure 800, in step 1924. Both the Task Management Module 540 and the Window Management Module 532 may create and manage copies of the window stack. These copies of the window stack can be synchronized or kept similar through communications between the Window Management Module 532 and the Task Management Module 540. Thus,

the Window Management Module 532 and the Task Management Module 540, based on the information determined by the Multi-Display Management Module 524, can assign dimensions 808, a stack position identifier 812 (e.g., window 1 1782, window 4 1770, etc.), a display identifier 816 (e.g., touch sensitive display 1 110, touch sensitive display 2 114, composite display identifier, etc,), and an active indicator 820, which is generally always set when the window is at the "top" of the stack. The logical data structure 800 may then be stored by both the Window Management Module 532 and the Task Management Module 540. Further, the Window Management Module 532 and the Task Management Module 540may thereinafrer manage the window stack and the logical data structure(s) 800.

[0118] FIG. 12 depicts a further window stacking configuration. A plurality of windows 1, 2, 3, 4, 5, 6, 7, and 8, whether from the same or different multi-screen or single-screen applications, are depicted. The touch sensitive display 110 currently has window 4 in the active display position while touch sensitive display 114 currently has window 5 in the active display position. The stack for touch sensitive display 110, from top to bottom, has window 4 in the active display position and windows 3, 2, and 1 positioned behind it. The stack for touch sensitive display 114, from top to bottom, has window 5 in the active display position and windows 6, 7, and 8 positioned behind it.

[0119] The desktops D1, D2, D3, D4, D5, and D6 are positioned behind the window stacks. The desktops can be viewed as a desktop stack different from the window stack. Viewed in this manner, the touch sensitive display 110 has a corresponding desktop stack comprising desktops D3, D2, and D1,with desktop D1 being at the bottom 2300 stack position and with desktop D3 at the top stack position being capable of being displayed with window 4 (depending on the window position and size (whether maximized or minimized), and the touch sensitive display 114 has a corresponding desktop stack has a corresponding desktop stack comprising desktops D4, D5, and D6,with desktop D6 being at the bottom 2304 stack position and with desktop D4 at the top stack position being capable of being displayed with window 5 (depending on the window position and size (whether maximized or minimized). Conceptually, the desktops can be viewed as a canvas divided, in this example, into six sections of which two can be displayed, at any one time, the touch sensitive displays 110, 114. When the device 100 is in the closed state, this conceptual model, in one configuration, persists. In this configuration, only one window and desktop stacks can be seen (that corresponding to the primary screen) but the other

window and desktop stacks are virtual; that is, they are maintained in memory but cannot be seen because the secondary screen is not enabled.

[0120] Single-screen application are typically related only to one window while multi-screen applications can be related to one or more windows. The latter hierarchical relationship is depicted in Fig. 8. As shown in Fig. 8, first window 1.1 is a root window, second window 1.1.1 is a node window, and third window 1.1.1.1 is a leaf window. The windows are in a hierarchical tree-like sequence and are controlled by a common multi-screen application. The root window 1.1 may be a top level view of the hierarchical application window sequence such that there is no parent window corresponding to the root window. The root window may be a parent to a node window. One or more node windows may be provided that are related as parent/children. A node window may also serve as a leaf window. By leaf window, it is meant that the leaf window has no corresponding node screen for which the leaf window is a parent. As such, the leaf window does not have any children node windows, Windows 1.1.1.1 and 1.1.2 are examples of leaf windows. The root view refers to all of the windows controlled by a multi-screen application at a point in time.

[0121] The "exposé" mode, which can be activated or deactivated by a user command, allows a user to quickly locate an open window, or to hide all windows and show the desktop without the need to click through many windows to find a specific target. The exposé mode can include one or more of the following four features for organizing windows. The "all windows" feature shows all open and unhidden windows, shrinking their appearance so they all fit on a single screen or display (when minimized) or simultaneously on multiple displays (when maximized). The "application windows" feature shows, on a single screen or display (when minimized) or simultaneously on multiple displays (when maximized), all open and unhidden windows for the currently active application. Again, the windows shrink to appear on the screen or display together, but generally they shrink less than the all windows feature because there are fewer windows in a single active or executing application compared to the system as a whole (which would include multiple active single- and/or multi-screen applications). During either the "all windows" or "application windows" modes, the user can cycle through windows of different applications by a suitable command, such as by clicking on the windows or selecting the window with an arrow key and pressing the enter key. When the exposé mode deactivates or is disabled, the selected window remains in the foreground. The "desktop" feature moves all windows off

the screen, with just the edges of the windows visible at the side of the screen, giving the user clear access to the desktop. The "mission control" feature gives an overview of all running applications just like "all windows" on a single screen or display (when minimized) or simultaneously on multiple displays (when maximized) but groups windows from the same application. At the top of the screen(s) or display(s) it gives quick access to the dashboard, spaces, and running full screen applications.

[0122] A number of examples will now be discussed with reference to closing windows and/or quitting executing or active applications. In the examples, a primary or main (minimized) exposé view in the exposé mode is denoted by "2+" because the exposé view contains multiple (two or more) active windows that are currently open while a secondary (or maximized) exposé view is denoted by "e". Either view, for example, is a display from one or more of the exposé mode features. The windows in the primary or secondary exposé view currently occupy inactive display positions in a respective stack.

[0123] Prior to discussing the examples, some fundamental rules regarding maximizing and minimizing displayed images will be discussed. A maximize or minimize operation is typically initiated by a user gesture, particularly a flip or drag. In a drag gesture, a 1:1 map reaction between the user's digit and the displayed image is used as the user "drags" the displayed image to the other touch sensitive display. The user typically releases the displayed image before it is fully moved to the other touch sensitive display. The initial speed after release is generally decreased progressively or gradually until the displayed image is fully located in the desired position. Too little displacement of the displayed image generally will not cause the image to be relocated. Such a low level of displacement is assumed to be accidental or incidental to the user using the device.

[0124] A multi-screen application window generally opens in minimized mode. A continuing drag and release gesture when a window of a multi-screen application, originally minimized in a first touch sensitive display, is at least half-way across the second touch sensitive display will generally cause the window to be maximized, or displayed in both the first and second touch sensitive displays. Dragging a window of a multi-screen application originally minimized in a first touch sensitive display to a second touch sensitive display all the way across to the far edge of the second touch sensitive display will generally maximize the window across both the first and second touch sensitive displays. When the user continues to drag in the same direction after

the window of the multi-screen application is expanded fully (or maximized across both the first and second touch sensitive displays and touching both far (non-adjacent) edges of the first and second touch sensitive displays), the window will pull away ftom the far edge of the first touch sensitive display and, once the far edge of the window is half-way across the first touch sensitive display, the application window will generally reposition in minimized mode in the second touch sensitive display.

[0125] The flick gesture will not generally maximize a window of a multi-screen application but generally will reposition a minimized window of a single- or multi-screen application. The flick gesture will generally not minimize window(s) of a maximized application. The flick gesture is a way to move rapidly through the windows of a stack, whether the windows are of a single- or multi-screen application.

[0126] A first example will be discussed with reference to Figs. 13A-C in which the device 100 is in dual portrait display mode. The first touch sensitive display 110 displays a minimized main view 2+, and the second touch sensitive display 114 displays a second desktop D2. The exposé view is minimized in that it occupies only one of the primary and secondary screens 104 and 108. In a gesture capture region 120 or 124 (or in other configurations the touch sensitive display 110 or 114) receives, ftom the user, a gesture 1300, such as a flick or drag gesture to the right (though one or more other gestures may be employed). By the gesture 1300, the user seeks to maximize the exposé view "e" by displaying the secondary expose view e on the right display, or touch sensitive display 114. As shown by Fig. 13B, the secondary exposé view is typically on the right side of the corresponding main exposé view 2+. Thus, the main exposé view 2+ remains on the left while the secondary exposé view 1304 moves out ftom "behind" the main exposé view 2+ to occupy the touch sensitive display 114 as shown in Fig. 13C. Stated another way, the secondary exposé view 1304 moves to occupy the active display position in the touch sensitive display 114, and the second desktop D2 occupies an inactive display position. In other embodiments, secondary exposé views are maintained on the left side of the corresponding main exposé view 2+.

[0127] A second example will be applied to Figs. 14A-C. The first touch sensitive and second touch sensitive displays 110, 114 are oppositely configured to Fig. 13A. Stated another way, the touch sensitive display 110 displays a first desktop D1, and the touch sensitive display 114 displays the corresponding main exposé view 2+. In a gesture capture region 120 or 124 (or in

other configurations the touch sensitive display 110 or 114) receives, from the user, a gesture 1300. Exemplary gestures include a drag or flick to the left. By the gesture 1300, the user seeks to maximize the exposé view by displaying the secondary exposé view to the right, or on touch sensitive display 114. As shown by Fig. 14B, when maximizing an exposé view to the right or on touch sensitive display 114, the corresponding main exposé view 2+ moves to the left, or towards the touch sensitive display 110. Thus, the main exposé view 2+ moves to active display position on the touch sensitive display 110, and the first desktop D1 to an inactive display position on the touch sensitive display 110 as shown by Fig. 14C. The primary exposé view does this by sliding itself over to the touch sensitive display 110, thereby revealing or uncovering the secondary exposé view as the primary exposé view moves. Conversely, the secondary exposé view 1304 moves to the active display position on the touch sensitive display 114, and the second desktop D2 (not shown) remains in an inactive display position. In other embodiments, the secondary exposé view moves to the left, or towards touch sensitive display 110 to occupy an active display position while the corresponding main exposé view 2+ remains on the right, or on the touch sensitive display 114.

[0128] With reference to Figs. 15A-C, a further example will be discussed. The first touch sensitive and second touch sensitive displays 110, 114 are configured the same as Fig. 13A. Stated another way, the touch sensitive display 114 displays a second desktop D2, and the touch sensitive display 110 displays the corresponding main exposé view 2+. In a gesture capture region 120 or 124 (or in other configurations the touch sensitive display 110 or 114) receives, ftom the user, a gesture 1300. Exemplary gestures include a drag or flick to the right. By the gesture 1300, the user seeks to move the main exposé view 2+ 1500 to the other touch sensitive display 114. As shown by Fig. 15B, the main exposé view 2+ 1500 moves ftom the touch sensitive display 110 to the touch sensitive display 114 to produce the final configuration shown in FIG. 15C. Stated another way, as a result of the gesture 1300 the main exposé view 2+ 1500 now occupies the active display position on touch sensitive display 114, the second desktop D2 now occupies an inactive display position relative to touch sensitive display 114, and the first desktop D1 now occupies an active display position on touch sensitive display 110.

[0129] A further example will now be discussed with reference to Figs. 16A-C. The first touch sensitive display 110 displays the main exposé view 2+ 1500, and the second touch sensitive display 114 displays the secondary exposé view e 1304. A user, by gesture 1300 (such as a drag

or flick gesture) to the right in a gesture capture region 120 or 124 (or in other configurations the touch sensitive display 110 or 114), seeks to minimize the main and secondary exposé views to the right to reveal only the main exposé view 2+ 1500. The main and secondary exposé views will be minimized in that they occupy only one of the primary and secondary screens 104 and 108. As a general rule, a secondary exposé view e 1304 does not cover the main exposé view 2+ 1500. As shown by Fig. 16B, minimizing an exposé view to the right moves the main exposé view 2+ 1500 to the right. As shown by Fig. 16C, the main exposé view 2+ 1500 now occupies the active display position in the touch sensitive display 114 while the first desktop D1 occupies the active display position in the touch sensitive display 110. The secondary exposé view e 1304 has moved to occupy an inactive display position with respect to the touch sensitive display 114.

[0130] A further example will now be discussed with reference to Figs. 17A-C. The first touch sensitive display 110 displays the main exposé view 2+ 1500, and the second touch sensitive display 114 displays the secondary exposé view e 1304. A user, by gesture 1300 (such as a drag or flick gesture) to the left in a gesture capture region 120 or 124 (or in other configurations the touch sensitive display 110 or 114), seeks to minimize the main and secondary exposé views to the left to reveal only the main exposé view 2+ 1500. As shown by Fig. 17B, minimizing an exposé view to the left moves the secondary exposé view e 1304 to the left, such that it appears to "slide under" the main exposé view 2+ 1500. As shown by Fig. 17C, the main exposé view 2+ 1500 remains in the active display position in the touch sensitive display 114 while the second desktop D2 occupies the active display position in the touch sensitive display 114. The secondary exposé view e 1304 has moved to the left to occupy an inactive display position relative to the touch sensitive display 110.

[0131] A further example will be discussed with reference to Figs. 18A-C in which the device 100 is in dual landscape display mode. In landscape display mode, exposé views behave similar to their behavior in portrait display mode. After maximization, the secondary exposé view e 1304 is typically maintained on the bottom touch sensitive display, which in Figs. 18A-C is depicted as touch sensitive display 114, while the primary or main exposé view 2+ 1500 is maintained on the upper touch sensitive display, which in Figs. 18A-C is depicted as touch sensitive display 110. Referring to Fig. 18A, the first touch sensitive display 110 displays a minimized main exposé view 2+ 1500, and the second touch sensitive display 114 displays a second desktop D2. The main exposé view 2+ 1500 is minimized in that it occupies only one of

the primary and secondary screens 104 and 108. In a gesture capture region 120 or 124 (or in other configurations the touch sensitive display 110 or 114) receives, from the user, a gesture 1300, such as a flick or drag gesture downwards (though one or more other gestures may be employed). By the gesture 1300, the user seeks to maximize the exposé view by displaying the secondary exposé view "e" 1304 on the lower touch sensitive display 114. As shown by Fig. 18B, the secondary exposé view e 1304 is typically below the corresponding main exposé view 2+ 1500. Thus, the main exposé view 2+ 1500 remains on the upper touch sensitive display 110 while the secondary exposé view e 1304 moves out from "behind" the main exposé view 2+ 1500 to occupy the touch sensitive display 114 as shown in Figs. 18B-C. Stated another way, the secondary exposé view e 1304 moves to occupy the active display position in the touch sensitive display 114, while the main exposé view 2+ 1500 remains in the active display position in the touch sensitive display 110 while the second desktop D2 occupies an inactive display position on the touch sensitive display 114. In other embodiments, secondary exposé views are maintained on the upper touch sensitive display 110 and the main exposé view 2+ 1500 is maintained in the lower touch sensitive display 114.

[0132] A further example will be discussed with reference to Figs. 19A-C in which the device 100 is in dual landscape display mode. Referring to Fig. 19A, the first touch sensitive display 110 displays a first desktop D1, and the second touch sensitive display 114 displays a minimized main exposé view 2+ 1500. In a gesture capture region 120 or 124 (or in other configurations the touch sensitive display 110 or 114) receives, from the user, a gesture 1300, such as a flick or drag gesture upwards (though one or more other gestures may be employed). By the gesture 1300, the user seeks to maximize the exposé view by displaying the secondary exposé view "e" 1304 on the lower touch sensitive display 114. As shown by Fig. 19B, the secondary exposé view e 1304 is typically below the corresponding main exposé view 2+ 1500. Thus, the main exposé view 2+ will move to the upper touch sensitive display 110 while the secondary exposé view e 1304 remains on the lower touch sensitive display 114 as shown in Figs. 19B-C. Stated another way, the main exposé view 2+ 1500 occupies the active display position in the upper touch sensitive display 110 while the secondary exposé view e 1304 occupies the active display position in the lower touch sensitive display 114. The first desktop D1, meanwhile, moves to an inactive display position with respect to the upper touch sensitive display 110. In other

embodiments, secondary exposé views are maintained on the upper touch sensitive display 110 and the main exposé view 2+ 1500 is maintained in the lower touch sensitive display 114.

[0133] With reference to Fig. 20A, the first touch sensitive and second touch sensitive displays 110, 114 are configured the same as Fig. 18A. Stated another way, the touch sensitive display 110 displays a first desktop D1, and the touch sensitive display 114 displays the (minimized) main exposé view 2+ 1500. In a gesture capture region 120 or 124 (or in other configurations the touch sensitive display 110 or 114) receives, from the user, a gesture 1300 in an upwardly direction. Exemplary gestures include a drag or flick to the right. By the gesture 1300, the user seeks to move the main exposé view 2+ 1500 from the lower touch sensitive display 114 to the upper touch sensitive display 110. As shown by Fig. 20B, the main exposé view 2+ 1500 moves from the touch sensitive display 114 to the touch sensitive display 110 to produce the final configuration shown in FIG. 20C. Stated another way, as a result of the gesture 1300 the main exposé view 2+ 1500 now occupies the active display position on touch sensitive display 110, the second desktop D2 now occupies an active display position relative to touch sensitive display 114, and the first desktop D1 now occupies an inactive display position on touch sensitive display 110.

[0134] A further example will now be discussed with reference to Figs. 21A-C. The upper touch sensitive display 110 displays the main exposé view 2+ 1500, and the lower touch sensitive display 114 displays the secondary exposé view e 1304. A user, by downward gesture 1300 (such as a drag or flick gesture) in a gesture capture region 120 or 124 (or in other configurations the touch sensitive display 110 or 114), seeks to minimize the main and secondary exposé views 1500, 1304 to the lower touch sensitive display 114 to reveal only the main exposé view 2+ 1500. The main and secondary exposé views 1500, 1304 will be minimized in that they occupy only one of the primary and secondary screens 104 and 108. As a general rule, a secondary exposé view e 1304 does not cover the main exposé view 2+ 1500 or is not displayed alone. As shown by Fig. 21B, minimizing an exposé view downwards moves the main exposé view 2+ 1500 from the upper touch sensitive display 110 to the lower touch sensitive display 114. As shown by Fig. 21C, the main exposé view 2+ 1500 now occupies the active display position in the lower touch sensitive display 114 while the first desktop D1 occupies the active display position in the upper touch sensitive display 110. The secondary exposé view e 1304 has

moved to occupy an inactive display position with respect to the lower touch sensitive display 114.

[0135] A further example will now be discussed with reference to Figs. 22A-C. The first touch sensitive display 110 displays the main exposé view 2+ 1500, and the second touch sensitive display 114 displays the secondary exposé view e 1304. A user, by upward gesture 1300 (such as a drag or flick gesture) in a gesture capture region 120 or 124 (or in other configurations the touch sensitive display 110 or 114), seeks to minimize the main and secondary exposé views to the upper touch sensitive display 110 to reveal only the main exposé view 2+ 1500. As shown by Fig. 22B, minimizing an exposé view in this manner moves the secondary exposé view e 1304 upwardly so that it "slides under" the main exposé view 2+ 1500. As shown by Fig. 22C, the main exposé view 2+ 1500 remains in the active display position in the touch sensitive display 110 while the second desktop D2 occupies the active display position in the touch sensitive display 114. The secondary exposé view e 1304 has moved to an inactive display position relative to the touch sensitive display 110.

[0136] In the various examples, middleware 520, particularly one or more of the Multi-Display Management (MDM) class 524, a Surface Cache class 528, a Window Management class 532, an Activity Management class 536, and an Application Management class 540, independently or collectively, receive a command to manipulate an exposé view in a predetermined manner, such as by detecting a gesture 1300 (step 2300). In response, middleware 520 determines whether the exposé view is currently maximized (in which case each of the touch sensitive displays 110 and 114 display one of the primary and secondary exposé views) or minimized (in which case only one of the touch sensitive displays displays an exposé views, which is the primary exposé view 1500) (step 2304). In decision diamond 2308, the middleware 520 decides how to process the command based on whether the exposé view is maximized or minimized. When the exposé view is maximized, the middleware 520 determines whether the device 100 is in the portrait or landscape display mode and, when in the landscape display mode, which of the touch sensitive displays 110 and 114 is on top and which is on the bottom (step 2312). When in portrait display mode, the middleware 520 causes the secondary exposé view 1304 to be displayed on a predetermined one of the left or right touch sensitive display. When in landscape display mode, the middleware 520 causes the secondary exposé view 1304 to be displayed on a predetermined one of the upper or lower touch sensitive display. When the exposé view is minimized, the

middleware 520 determines whether the device 100 is in the portrait or landscape display mode and, when in the landscape display mode, which of the touch sensitive displays 110 and 114 is on top and which is on the bottom (step 2316). When in portrait display mode, the middleware 520 causes the secondary exposé view 1304 to no longer be displayed on a predetermined one of the left or right touch sensitive display. When in landscape display mode, the middleware 520 causes the secondary exposé view 1304 to no longer be displayed on a predetermined one of the upper or lower touch sensitive display

[0137] The exemplary systems and methods of this disclosure have been described in relation to communication devices. However, to avoid unnecessarily obscuring the present disclosure, the preceding description omits a number of known structures and devices. This omission is not to be construed as a limitation of the scopes of the claims. Specific details are set forth to provide an understanding of the present disclosure. It should however be appreciated that the present disclosure may be practiced in a variety of ways beyond the specific detail set forth herein.

[0138] Furthermore, while the exemplary aspects, embodiments, and/or configurations illustrated herein show the various components of the system collocated, certain components of the system can be located remotely, at distant portions of a distributed network, such as a LAN and/or the Internet, or within a dedicated system. Thus, it should be appreciated, that the components of the system can be combined in to one or more devices, such as a communication device, or collocated on a particular node of a distributed network, such as an analog and/or digital telecommunications network, a packet-switch network, or a circuit-switched network. It will be appreciated from the preceding description, and for reasons of computational efficiency, that the components of the system can be arranged at any location within a distributed network of components without affecting the operation of the system. For example, the various components can be located in a switch such as a PBX and media server, gateway, in one or more communications devices, at one or more users' premises, or some combination thereof. Similarly, one or more functional portions of the system could be distributed between a telecommunications device(s) and an associated computing device.

[0139] Furthermore, it should be appreciated that the various links connecting the elements can be wired or wireless links, or any combination thereof, or any other known or later developed element(s) that is capable of supplying and/or communicating data to and from the connected

elements. These wired or wireless links can also be secure links and may be capable of communicating encrypted information. Transmission media used as links, for example, can be any suitable carrier for electrical signals, including coaxial cables, copper wire and fiber optics, and may take the form of acoustic or light waves, such as those generated during radio-wave and infra-red data communications.

[0140] Also, while the flowcharts have been discussed and illustrated in relation to a particular sequence of events, it should be appreciated that changes, additions, and omissions to this sequence can occur without materially affecting the operation of the disclosed embodiments, configuration, and aspects.

[0141] A number of variations and modifications of the disclosure can be used. It would be possible to provide for some features of the disclosure without providing others.

[0142] In other embodiments, other rules may be applied. For example, the preferred orientations of the primary and secondary exposé views is different from the orientations described above.

[0143] In yet another embodiment, the systems and methods of this disclosure can be implemented in conjunction with a special purpose computer, a programmed microprocessor or microcontroller and peripheral integrated circuit element(s), an ASIC or other integrated circuit, a digital signal processor, a hard-wired electronic or logic circuit such as discrete element circuit, a programmable logic device or gate array such as PLD, PLA, FPGA, PAL, special purpose computer, any comparable means, or the like. In general, any device(s) or means capable of implementing the methodology illustrated herein can be used to implement the various aspects of this disclosure. Exemplary hardware that can be used for the disclosed embodiments, configurations and aspects includes computers, handheld devices, telephones (e.g., cellular, Internet enabled, digital, analog, hybrids, and others), and other hardware known in the art. Some of these devices include processors (e.g., a single or multiple microprocessors), memory, nonvolatile storage, input devices, and output devices. Furthermore, alternative software implementations including, but not limited to, distributed processing or component/object distributed processing, parallel processing, or virtual machine processing can also be constructed to implement the methods described herein.

[0144] In yet another embodiment, the disclosed methods may be readily implemented in conjunction with software using object or object-oriented software development environments

that provide portable source code that can be used on a variety of computer or workstation platforms. Alternatively, the disclosed system may be implemented partially or fully in hardware using standard logic circuits or VLSI design. Whether software or hardware is used to implement the systems in accordance with this disclosure is dependent on the speed and/or efficiency requirements of the system, the particular function, and the particular software or hardware systems or microprocessor or microcomputer systems being utilized.

[0145] In yet another embodiment, the disclosed methods may be partially implemented in software that can be stored on a storage medium, executed on programmed general-purpose computer with the cooperation of a controller and memory, a special purpose computer, a microprocessor, or the like. In these instances, the systems and methods of this disclosure can be implemented as program embedded on personal computer such as an applet, JAVA® or CGI script, as a resource residing on a server or computer workstation, as a routine embedded in a dedicated measurement system, system component, or the like. The system can also be implemented by physically incorporating the system and/or method into a software and/or hardware system.

[0146] Although the present disclosure describes components and functions implemented in the aspects, embodiments, and/or configurations with reference to particular standards and protocols, the aspects, embodiments, and/or configurations are not limited to such standards and protocols. Other similar standards and protocols not mentioned herein are in existence and are considered to be included in the present disclosure. Moreover, the standards and protocols mentioned herein and other similar standards and protocols not mentioned herein are periodically superseded by faster or more effective equivalents having essentially the same functions. Such replacement standards and protocols having the same functions are considered equivalents included in the present disclosure.

[0147] The present disclosure, in various aspects, embodiments, and/or configurations, includes components, methods, processes, systems and/or apparatus substantially as depicted and described herein, including various aspects, embodiments, configurations embodiments, subcombinations, and/or subsets thereof. Those of skill in the art will understand how to make and use the disclosed aspects, embodiments, and/or configurations afrer understanding the present disclosure. The present disclosure, in various aspects, embodiments, and/or configurations, includes providing devices and processes in the absence of items not depicted

and/or described herein or in various aspects, embodiments, and/or configurations hereof, including in the absence of such items as may have been used in previous devices or processes, e.g., for improving performance, achieving ease and\or reducing cost of implementation.

[0148] The foregoing discussion has been presented for purposes of illustration and description. The foregoing is not intended to limit the disclosure to the form or forms disclosed herein. In the foregoing Detailed Description for example, various features of the disclosure are grouped together in one or more aspects, embodiments, and/or configurations for the purpose of streamlining the disclosure. The features of the aspects, embodiments, and/or configurations of the disclosure may be combined in alternate aspects, embodiments, and/or configurations other than those discussed above. This method of disclosure is not to be interpreted as reflecting an intention that the claims require more features than are expressly recited in each claim. Rather, as the following claims reflect, inventive aspects lie in less than all features of a single foregoing disclosed aspect, embodiment, and/or configuration. Thus, the following claims are hereby incorporated into this Detailed Description, with each claim standing on its own as a separate preferred embodiment of the disclosure.

[0149] Moreover, though the description has included description of one or more aspects, embodiments, and/or configurations and certain variations and modifications, other variations, combinations, and modifications are within the scope of the disclosure, e.g., as may be within the skill and knowledge of those in the art, after understanding the present disclosure. It is intended to obtain rights which include alternative aspects, embodiments, and/or configurations to the extent permitted, including alternate, interchangeable and/or equivalent structures, functions, ranges or steps to those claimed, whether or not such alternate, interchangeable and/or equivalent structures, functions, ranges or steps are disclosed herein, and without intending to publicly dedicate any patentable subject matter.

What is claimed is:

1. A method, comprising:

receiving, by processor executable middleware in a multi-display communication device, a command to at least one of minimize and maximize an exposé view, the exposé view comprising a primary exposé view and a secondary exposé view at least one of displayed and to be displayed by different touch sensitive displays;

applying, by the processor executable middleware, the following rules:

(B1) when the exposé view is to be minimized, selecting a predetermined touch sensitive display for displaying the primary exposé view and ceasing to display the secondary exposé view; and

(B2) when the exposé view is to bemaximized, selecting a predetermined touch sensitive display for displaying the primary exposé view and causing the secondary exposé view to be displayed on the other touch sensitive display.

2. The method of claim 1, wherein rule (B2) applies.

3. The method of claim 2, wherein the command is a drag or a flick, wherein the primary exposé view is in the predetermined touch sensitive display when the gesture is received, and wherein the secondary exposé view appears to move from under the primary exposé view in the predetermined touch sensitive display to the other touch sensitive display.

4. The method of claim 2, wherein the command is a drag or a flick, wherein the primary exposé view is not in the predetermined touch sensitive display when the gesture is received, wherein the primary exposé view moves from the other touch sensitive display to the predetermined touch sensitive display, thereby uncovering the secondary exposé view for display by the other touch sensitive display.

5. The method of claim 1, wherein the multi-display communication device is in the portrait display mode and wherein the predetermined touch sensitive display is the lefr-most touch sensitive display.

6. The method of claim 1, wherein the multi-display communication device is in the landscape display mode and wherein the predetermined touch sensitive display is the top-most touch sensitive display.

7. The method of claim 1, wherein the secondary exposé view is not permitted to be displayed in the predetermined touch sensitive display and wherein the secondary exposé view is not permitted to be displayed in the absence of concurrent display of the primary exposé view.

8. The method of claim 1, wherein the primary and secondary exposé views display at least one of the following:

all open and unhidden windows;

all open and unhidden windows for a selected currently active application; and

an overview of all currently active applications and groups windows from a common application.

9. The method of claim 1, wherein rule (B1) applies.

10. The method of claim 9, wherein the command is a drag or a flick, wherein the primary exposé view is in the predetermined touch sensitive display when the gesture is received, and wherein the secondary exposé view appears to move from the other touch sensitive display to slide behind the primary exposé view displayed in the predetermined touch sensitive display.

11. The method of claim 9, wherein the command is a drag or a flick, wherein the secondary exposé view is in the predetermined touch sensitive display when the gesture is received, and wherein the primary exposé view appears to move from the other touch sensitive display to cover the primary exposé view displayed in the predetermined touch sensitive display.

12. A non-transient and tangible computer readable medium comprising processor executable instructions to perform the steps of claim 1.

13. A communication device, comprising:

at least first and second touch sensitive displays to display windows;

a processor; and

a computer readable memory, the computer readable memory comprising processor executable middleware operable to:

receive a command to at least one of minimize and maximize an exposé view, the exposé view comprising a primary exposé view and a secondary exposé view at least one of displayed and to be displayed by different touch sensitive displays;

apply the following rules:

(i) when the exposé view is to be minimized, selecting a predetermined touch sensitive display for displaying the primary exposé view and ceasing to display the secondary exposé view on the other touch sensitive display; and

(ii) when the exposé view is to bemaximized, selecting a predetermined touch sensitive display for displaying the primary exposé view and causing the secondary exposé view to be displayed on the other touch sensitive display.

14. The device of claim 13, wherein rule (ii) applies.

15. The device of claim 14, wherein the command is a drag or a flick, wherein the primary exposé view is in the predetermined touch sensitive display when the gesture is received, and wherein the secondary exposé view appears to move from under the primary exposé view in the predetermined touch sensitive display to the other touch sensitive display.

16. The device of claim 14, wherein the command is a drag or a flick, wherein the primary exposé view is not in the predetermined touch sensitive display when the gesture is received, wherein the primary exposé view moves from the other touch sensitive display to the predetermined touch sensitive display, thereby uncovering the secondary exposé view for display by the other touch sensitive display.

17. The device of claim 13, wherein the multi-display communication device is in the portrait display mode and wherein the predetermined touch sensitive display is the left-most touch sensitive display.

18. The device of claim 13, wherein the multi-display communication device is in the landscape display mode and wherein the predetermined touch sensitive display is the top-most touch sensitive display.

19. The device of claim 13, wherein the secondary exposé view is not permitted to be displayed in the predetermined touch sensitive display and wherein the secondary exposé view is not permitted to be displayed in the absence of concurrent display of the primary exposé view.

20. The device of claim 13, wherein the primary and secondary exposé views display at least one of the following:

all open and unhidden windows;

all open and unhidden windows for a selected currently active application; and

an overview of all currently active applications and groups windows from a common application.

21. The device of claim 13, wherein rule (i) applies.

22. The device of claim 21, wherein the command is a drag or a flick, wherein the primary exposé view is in the predetermined touch sensitive display when the gesture is received, and wherein the secondary exposé view appears to move from the other touch sensitive display to slide behind the primary exposé view displayed in the predetermined touch sensitive display.

23. The device of claim 21, wherein the command is a drag or a flick, wherein the secondary exposé view is in the predetermined touch sensitive display when the gesture is received, and wherein the primary exposé view appears to move from the other touch sensitive display to cover the primary exposé view displayed in the predetermined touch sensitive display.

ABSTRACT

[0150] The present disclosure is directed to methodologies and devices for handling maximizing and minimizing of exposé views.

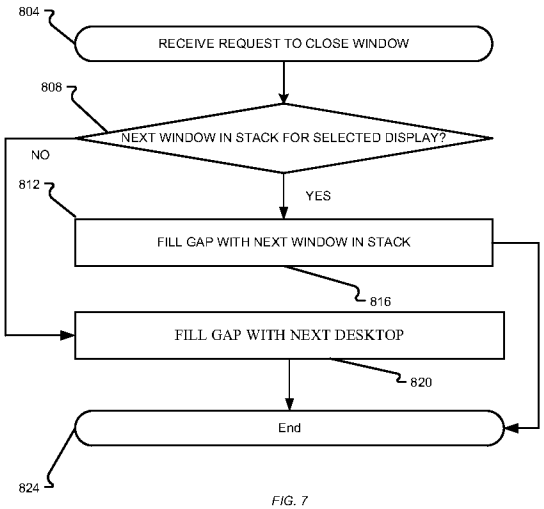


FIG. 7

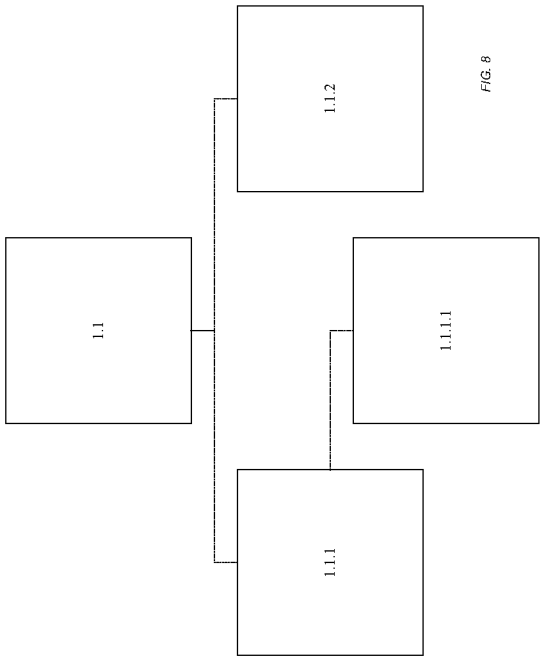


FIG. 8

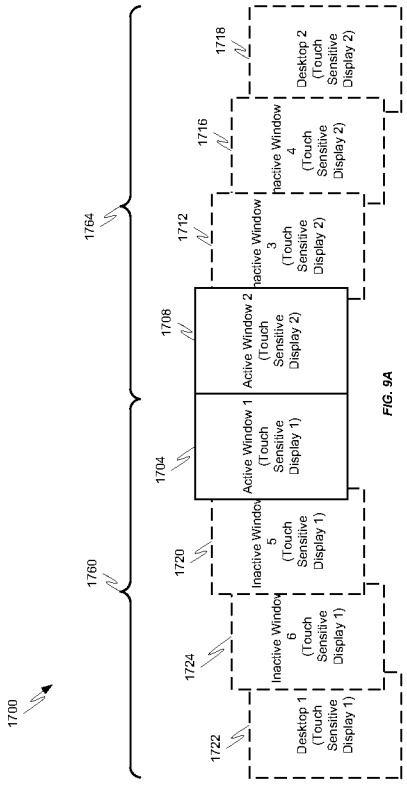


FIG. 9A

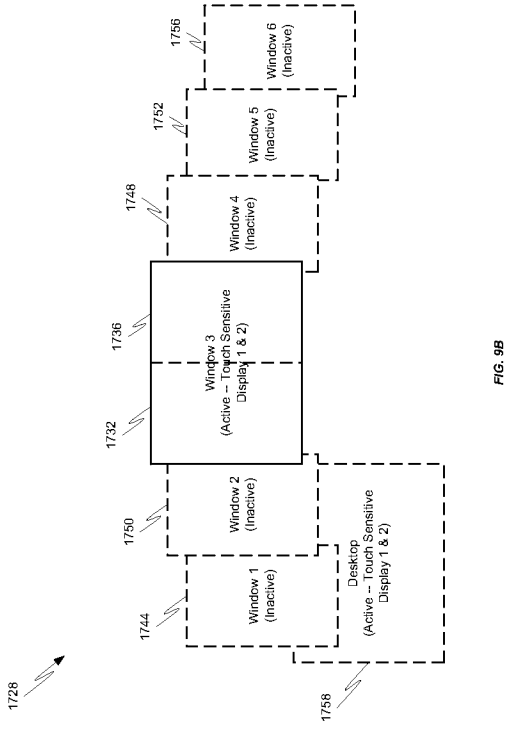


FIG. 9B

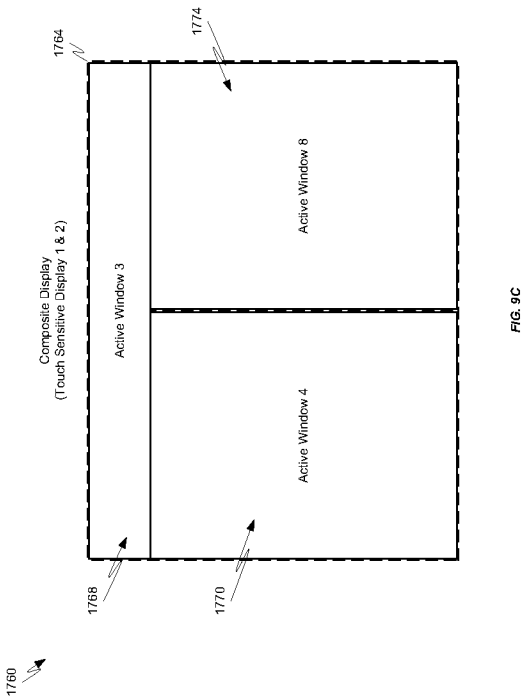


FIG. 9C

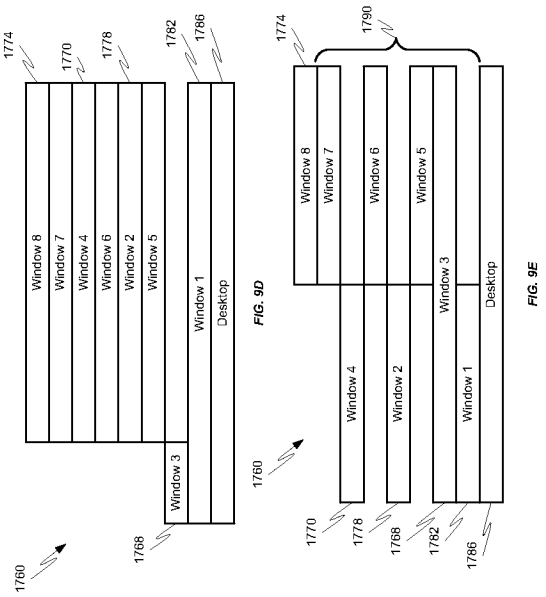


FIG. 9D

FIG. 9E

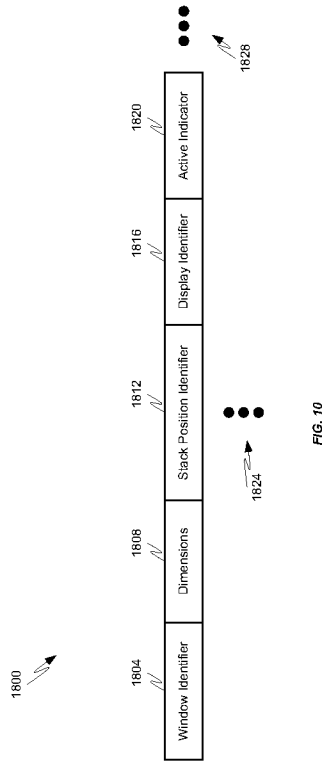


FIG. 10

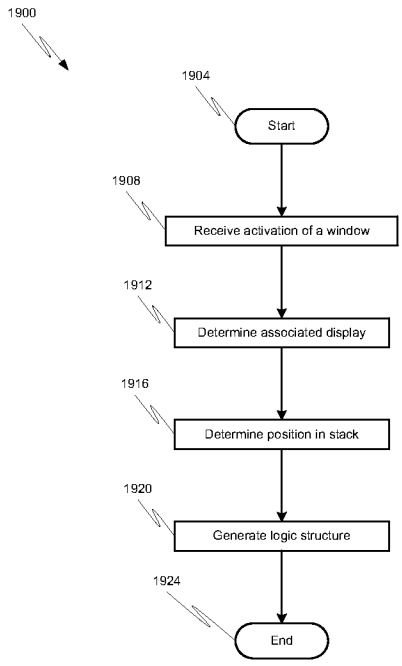


FIG. 11

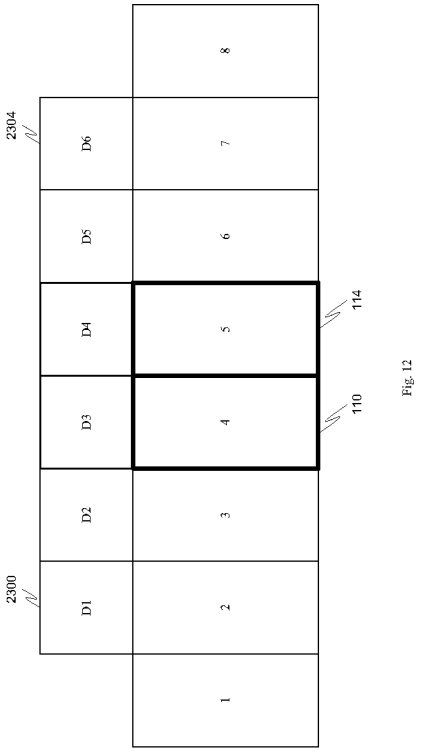


Fig. 12

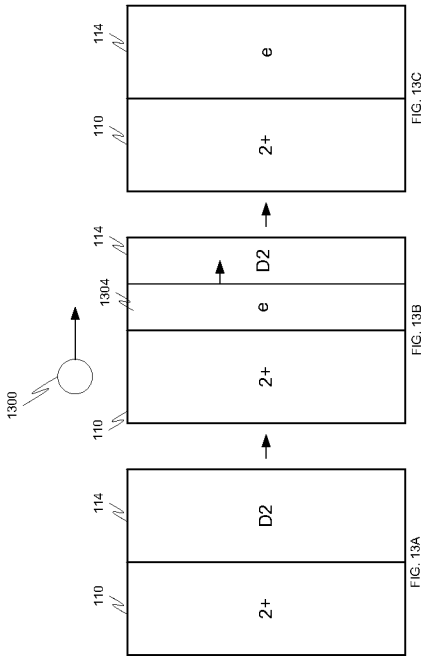


FIG. 13A FIG. 13B FIG. 13C

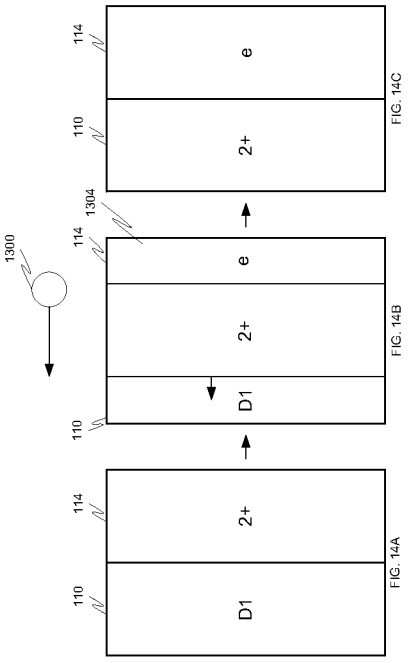


FIG. 14A FIG. 14B FIG. 14C

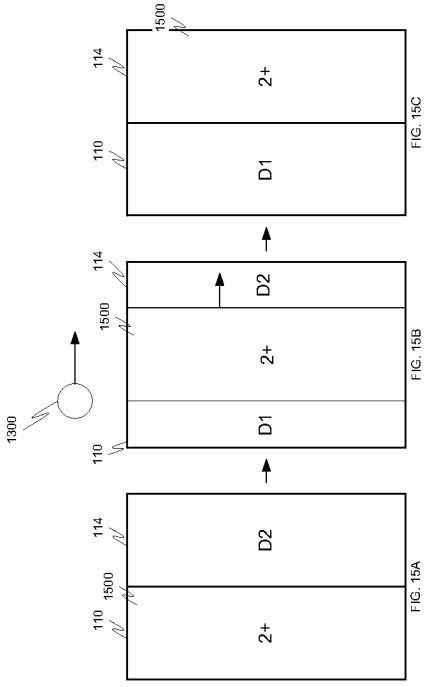
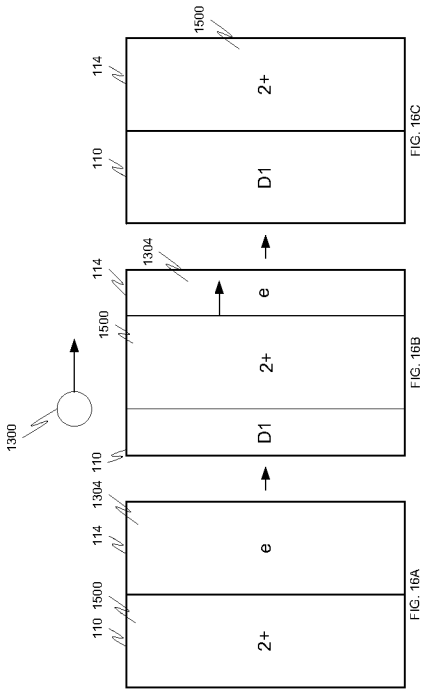
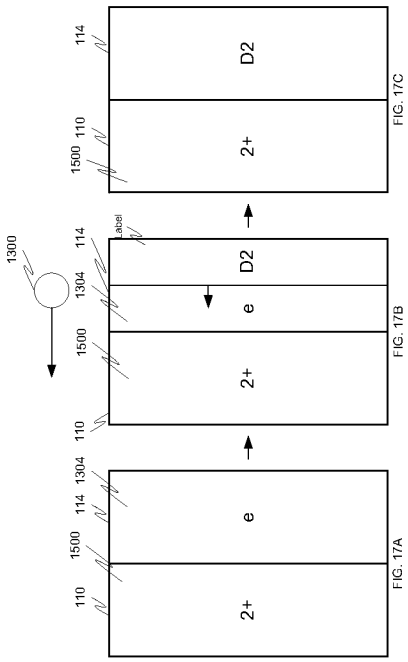
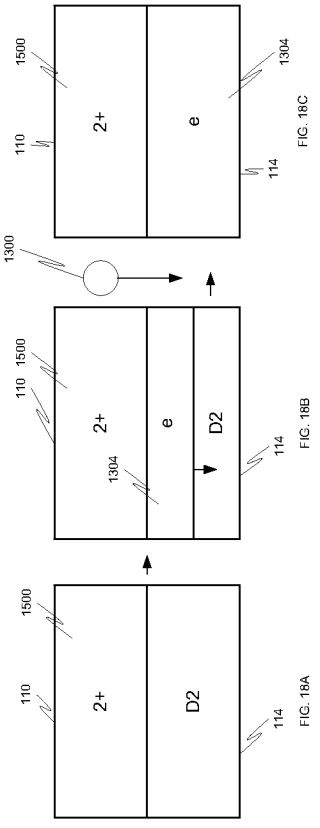
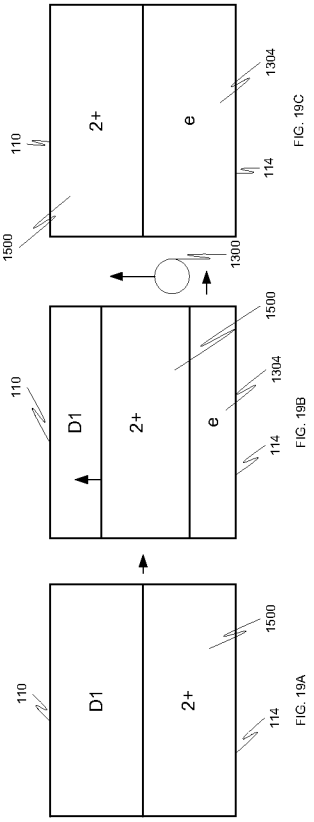


FIG. 15A FIG. 15B FIG. 15C


1500
114
2+
110
D1
FIG. 16C
1304
114
e
1500
2+
1300
110
D1
FIG. 16B
1304
114
e
1500
110
2+
FIG. 16A


114
D2
1500
110
2+
FIG. 17C
1304
label
114
D2
1300
e
1500
2+
110
FIG. 17B
1304
114
e
1500
110
2+
FIG. 17A


1500
110
2+
D2
114
FIG. 18A
1500
110
2+
e
D2
1304
114
FIG. 18B
1300
1500
110
2+
e
1304
114
FIG. 18C


110
D1
1500
2+
114
FIG. 19A
110
D1
2+
1500
e
1304
114
FIG. 19B
110
2+
e
1304
114
1500
1300
FIG. 19C

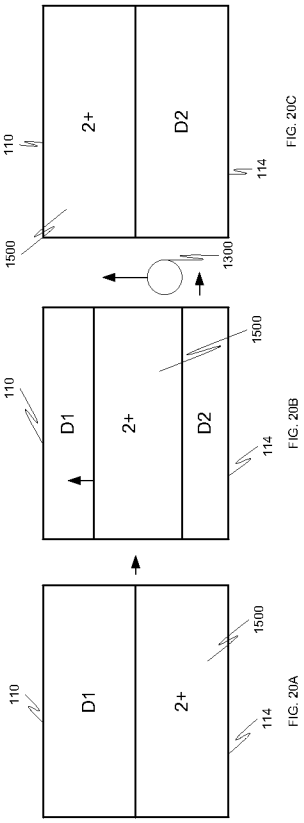


FIG. 20C

FIG. 20B

FIG. 20A

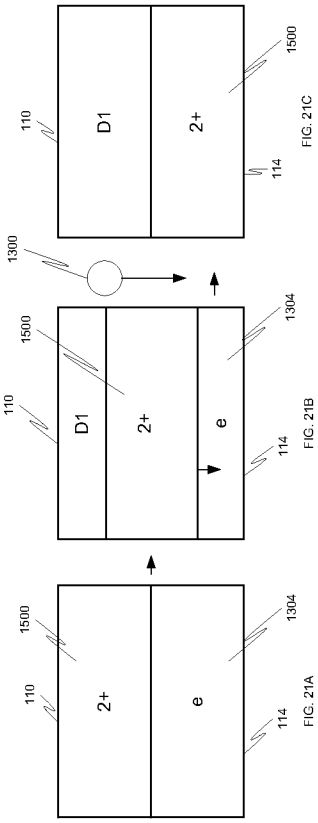


FIG. 21C

FIG. 21B

FIG. 21A

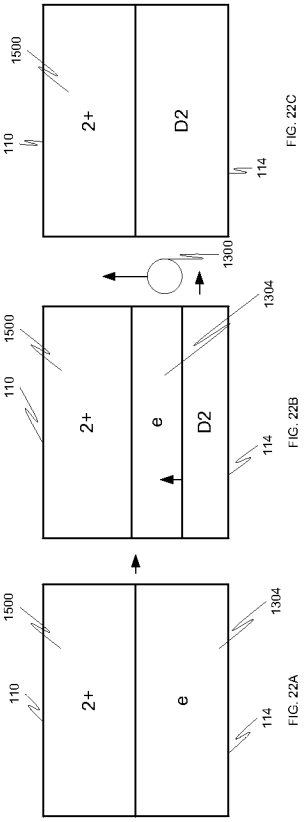


FIG. 22C

FIG. 22B

FIG. 22A

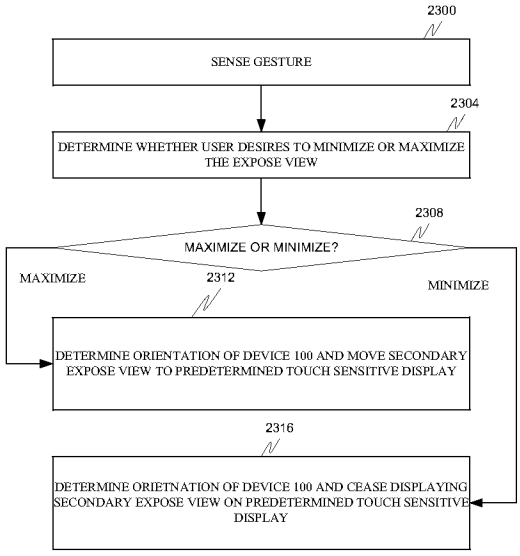


FIG. 23

LONG DRAG GESTURE IN USER INTERFACE

[0100] The device 100 manages desktops and/or windows with at least one window stack 1700, 1728, as shown in Figs. 12A and 12B. A window stack 1700, 1728 is a logical arrangement of active and/or inactive windows for a multi-screen device. For example, the window stack 1700, 1728 may be logically similar to a deck of cards, where one or more windows or desktops are arranged in order, as shown in Figs. 12A and 12B. An active window is a window that is currently being displayed on at least one of the touch sensitive displays 110, 114. For example, windows 104 and 108 are active windows and are displayed on touch sensitive displays 110 and 114. An inactive window is a window that was opened and displayed but is now "behind" an active window and not being displayed. In embodiments, an inactive window may be for an application that is suspended, and thus, the window is not displaying active content. For example, windows 1712, 1716, 1720, and 1724 are inactive windows.

[0101] A window stack 1700, 1728 may have various arrangements or organizational structures. In the embodiment shown in Fig. 12A, the device 100 includes a first stack 1760 associated with a first touch sensitive display 110 and a second stack associated with a second touch sensitive display 114. Thus, each touch sensitive display 110, 114 can have an associated window stack 1760, 1764. These two window stacks 1760, 1764 may have different numbers of windows arranged in the respective stacks 1760, 1764. Further, the two window stacks 1760, 1764 can also be identified differently and managed separately. Thus, the first window stack 1760 can be arranged in order from a first window 1704 to a next window 1720 to a last window 1724 and finally to a desktop 1722, which, in embodiments, is at the "bottom" of the window stack 1760. In embodiments, the desktop 1722 is not always at the "bottom" as application windows can be arranged in the window stack below the desktop 1722, and the desktop 1722 can be brought to the "top" of a stack over other windows during a desktop reveal. Likewise, the second stack 1764 can be arranged from a first window 1708 to a next window 1712 to a last window 1716, and finally to a desktop 1718, which, in embodiments, is a single desktop area, with desktop 1722, under all the windows in both window stack 1760 and window stack 1764. A logical data structure for managing the two window stacks 1760, 1764 may be as described in conjunction with Fig. 13.

[0102] Another arrangement for a window stack 1728 is shown in Fig. 12B. In this embodiment, there is a single window stack 1728 for both touch sensitive displays 110, 114. Thus, the window stack 1728 is arranged from a desktop 1758 to a first window 1744 to a last window 1756. A window can be arranged in a position among all windows without an association to a specific touch sensitive display 110, 114. In this embodiment, a window is in the order of windows. Further, at least one window is identified as being active. For example, a single window may be rendered in two portions 1732 and 1736 that are displayed on the first touch sensitive screen 110 and the second touch sensitive screen 114. The single window may only occupy a single position in the window stack 1728 although it is displayed on both displays 110, 114.

[0103] Yet another arrangement of a window stack 1760 is shown in Figs. 12C through 12E. The window stack 1760 is shown in three "elevation" views. In Fig. 12C, the top of the window stack 1760 is shown. Two sides of the window stack 1760 are shown in Figs. 12D and 12E. In this embodiment, the window stack 1760 resembles a stack of bricks. The windows are stacked on each other. Looking from the top of the window stack 1760 in Fig, 12C, only the top most windows in the window stack 1760 are seen in different portions of the composite display 1764. The composite display 1764 represents a logical model for the entire display area of the device 100, which can include touch sensitive display 110 and touch sensitive display 114. A desktop 1786 or a window can occupy part or all of the composite display 1764.

[0104] In the embodiment shown, the desktop 1786 is the lowest display or "brick" in the window stack 1760. Thereupon, window 1 1782, window 2 1782, window 3 1768, and window 4 1770 are layered. Window 1 1782, window 3 1768, window 2 1782, and window 4 1770 only occupy a portion of the composite display 1764. Thus, another part of the stack 1760 includes window 8 1774 and windows 5 through 7 shown in section 1790. Only the top window in any portion of the composite display 1764 is actually rendered and displayed. Thus, as shown in the top view in Fig. 12C, window 4 1770, window 8 1774, and window 3 1768 are displayed as being at the top of the display in different portions of the window stack 1760. A window can be dimensioned to occupy only a portion of the composite display 1760 to "reveal" windows lower in the window stack 1760. For example, window 3 1768 is lower

in the stack than both window 4 1770 and window 8 1774 but is still displayed. A logical data structure to manage the window stack can be as described in conjunction with Fig. 13.

[0105] When a new window is opened, the newly activated window is generally positioned at the top of the stack. However, where and how the window is positioned within the stack can be a function of the orientation of the device 100, the context of what programs, functions, software, etc. are being executed on the device 100, how the stack is positioned when the new window is opened, etc. To insert the window in the stack, the position in the stack for the window is determined and the touch sensitive display 110, 114 to which the window is associated may also be determined. With this information, a logical data structure for the window can be created and stored. When user interface or other events or tasks change the arrangement of windows, the window stack(s) can be changed to reflect the change in arrangement. It should be noted that these same concepts described above can be used to manage the one or more desktops for the device 100.

[0106] A logical data structure 1800 for managing the arrangement of windows or desktops in a window stack is shown in Fig. 13. The logical data structure 1800 can be any data structure used to store data whether an object, record, file, etc. The logical data structure 1800 can be stored in any type of database or data storage system, regardless of protocol or standard. In embodiments, the logical data structure 1800 includes one or more portions, fields, attributes, etc. that store data in a logical arrangement that allows for easy storage and retrieval of the information. Hereinafrer, these one or more portions, fields, attributes, etc. shall be described simply as fields. The fields can store data for a window identifier 1804, dimensions 1808, a stack position identifier 1812, a display identifier 1816, and/or an active indicator 1820. Each window in a window stack can have an associated logical data structure 1800. While only a single logical data structure 1800 is shown in Fig. 13, there may be more or fewer logical data structures 1800 used with a window stack (based on the number of windows or desktops in the stack), as represented by ellipses 1824. Further, there may be more or fewer fields than those shown in Fig. 13, as represented by ellipses 1828.

[0107] A window identifier 1804 can include any identifier (ID) that uniquely identifies the associated window in relation to other windows in the window stack. The window identifier 1804 can be a globally unique identifier (GUID), a numeric ID, an alphanumeric ID, or other type of identifier. In embodiments, the window identifier 1804 can be one, two, or any

number of digits based on the number of windows that can be opened. In alternative embodiments, the size of the window identifier 1804 may change based on the number of windows opened. While the window is open, the window identifier 1804 may be static and remain unchanged.

[0108] Dimensions 1808 can include dimensions for a window in the composite display 1760. For example, the dimensions 1808 can include coordinates for two or more corners of the window or may include one coordinate and dimensions for the width and height of the window. These dimensions 1808 can delineate what portion of the composite display 1760 the window may occupy, which may the entire composite display 1760 or only part of composite display 1760. For example, window 4 1770 may have dimensions 1880 that indicate that the window 1770 will occupy only part of the display area for composite display 1760, as shown in Figs. 12C through 12E. As windows are moved or inserted in the window stack, the dimensions 1808 may change.

[0109] A stack position identifier 1812 can be any identifier that can identify the position in the stack for the window or may be inferred from the window's control record within a data structure, such as a list or a stack. The stack position identifier 1812 can be a GUID, a numeric ID, an alphanumeric ID, or other type of identifier. Each window or desktop can include a stack position identifier 1812. For example, as shown in Fig. 12A, window 1 1704 in stack 1 1760 can have a stack position identifier 1812 of 1 identifying that window 1704 is the first window in the stack 1760 and the active window. Similarly, window 6 1724 can have a stack position identifier 1812 of 3 representing that window 1724 is the third window in the stack 1760. Window 2 1708 can also have a stack position identifier 1812 of 1 representing that window 1708 is the first window in the second stack 1764. As shown in Fig. 12B, window 1 1744 can have a stack position identifier 1812 of 1, window 3, rendered in portions 1732 and 1736, can have a stack position identifier 1812 of 3, and window 6 1756 can have a stack position identifier 1812 of 6. Thus, depending on the type of stack, the stack position identifier 1812 can represent a window's location in the stack.

[0110] A display identifier 1816 can identify that the window or desktop is associated with a particular display, such as the first display 110 or the second display 114, or the composite display 1760 composed of both displays. While this display identifier 1816 may not be needed for a multi-stack system, as shown in Fig. 12A, the display identifier 1816 can

indicate whether a window in the serial stack of Fig. 12B is displayed on a particular display. Thus, window 3 may have two portions 1732 and 1736 in Fig. 12B. The first portion 1732 may have a display identifier 1816 for the first display while the second portion 1736 may have a display identifier 1816 for the second display 114. However, in alternative embodiments, the window may have two display identifier 1816 that represent that the window is displayed on both of the displays 110, 114, or a display identifier 1816 identifying the composite display. In another alternate embodiment, the window may have a single display identifier 1816 to represent that the window is displayed on both of the displays 110, 114.

[0111] Similar to the display identifier 1816, an active indicator 1820 may not be needed with the dual stack system of Fig. 12A, as the window in stack position 1 is active and displayed. In the system of Fig. 12B, the active indicator 1820 can indicate which window(s) in the stack is being displayed. Thus, window 3 may have two portions 1732 and 1736 in Fig. 12. The first portion 1732 may have an active indicator 1820 while the second portion 1736 may also have an active indicator 1820. However, in alternative embodiments, window 3 may have a single active indicator 1820. The active indicator 1820 can be a simple flag or bit that represents that the window is active or displayed.

[0112] An embodiment of a method 1900 for creating a window stack is shown in Fig. 14. While a general order for the steps of the method 1900 is shown in Fig. 14. Generally, the method 1900 starts with a start operation 1904 and ends with an end operation 1928. The method 1900 can include more or fewer steps or can arrange the order of the steps differently than those shown in Fig. 14. The method 1900 can be executed as a set of computer-executable instructions executed by a computer system and encoded or stored on a computer readable medium. Hereinafter, the method 1900 shall be explained with reference to the systems, components, modules, software, data structures, user interfaces, etc. described in conjunction with Figs. 1-13.

[0113] A multi-screen device 100 can receive activation of a window, in step 1908. In embodiments, the multi-screen device 100 can receive activation of a window by receiving an input from the touch sensitive display 110 or 114, the configurable area 112 or 116, a gesture capture region 120 or 124, or some other hardware sensor operable to receive user interface inputs. The processor may execute the Task Management Module 540 may receive

the input. The Task Management Module 540 can interpret the input as requesting an application task to be executed that will open a window in the window stack.

[0114] In embodiments, the Task Management Module 540 places the user interface interaction in the task stack 552 to be acted upon by the Display Configuration Module 568 of the Multi-Display Management Module 524. Further, the Task Management Module 540 waits for information from the Multi-Display Management Module 524 to send instructions to the Window Management Module 532 to create the window in the window stack.

[0115] The Multi-Display Management Module 524, upon receiving instruction from the Task Management Module 540, determines to which touch portion of the composite display 1760, the newly activated window should be associated, in step 1912. For example, window 4 1770 is associated with the a portion of the composite display 1764 In embodiments, the device state module 574 of the Multi-Display Management Module 524 may determine how the device is oriented or in what state the device is in, e.g., open, closed, portrait, etc. Further, the preferences module 572 and/or requirements module 580 may determine how the window is to be displayed. The gesture module 576 may determine the user's intentions about how the window is to be opened based on the type of gesture and the location of where the gesture is made.

[0116] The Display Configuration Module 568 may use the input from these modules and evaluate the current window stack 1760 to determine the best place and the best dimensions, based on a visibility algorithm, to open the window. Thus, the Display Configuration Module 568 determines the best place to put the window at the top of the window stack 1760, in step 1916. The visibility algorithm, in embodiments, determines for all portions of the composite display, which windows are at the top of the stack. For example, the visibility algorithm determines that window 3 1768, window 4 1770, and window 8 1774 are at the top of the stack 1760 as viewed in Figs. 12C through 12E. Upon determining where to open the window, the Display Configuration Module 568 can assign a display identifier 816 and possibly dimensions 808 to the window. The display identifier 816 and dimensions 808 can then be sent back to the Task Management Module 540. The Task Management Module 540 may then assign the window a stack position identifier 812 indicating the windows position at the top of the window stack.

[0117] In embodiments, the Task Management Module 540 sends the window stack information and instructions to render the window to the Window Management Module 532. The Window Management Module 532 and the Task Management Module 540 can create the logical data structure 800, in step 1924. Both the Task Management Module 540 and the Window Management Module 532 may create and manage copies of the window stack. These copies of the window stack can be synchronized or kept similar through communications between the Window Management Module 532 and the Task Management Module 540. Thus, the Window Management Module 532 and the Task Management Module 540, based on the information determined by the Multi-Display Management Module 524, can assign dimensions 808, a stack position identifier 812 (e.g., window 1 1782, window 4 1770, etc.), a display identifier 816 (e.g., touch sensitive display 1 110, touch sensitive display 2 114, composite display identifier, etc,), and an active indicator 820, which is generally always set when the window is at the "top" of the stack. The logical data structure 800 may then be stored by both the Window Management Module 532 and the Task Management Module 540. Further, the Window Management Module 532 and the Task Management Module 540may thereinafter manage the window stack and the logical data structure(s) 800.

[0118] FIG. 15 depicts a further window stacking configuration. A plurality of windows 1, 2, 3, 4, 5, 6, 7, and 8, whether from the same or different multi-screen or single-screen applications, are depicted. The touch sensitive display 110 currently has window 4 in the active display position while touch sensitive display 114 currently has window 5 in the active display position. The stack for touch sensitive display 110, from top to bottom, has window 4 in the active display position and windows 3, 2, and 1 positioned behind it. The stack for touch sensitive display 114, from top to bottom, has window 5 in the active display position and windows 6, 7, and 8 positioned behind it.

[0119] The desktops D1, D2, D3, D4, D5, and D6 are positioned behind the window stacks. The desktops can be viewed as a desktop stack different from the window stack. Viewed in this manner, the touch sensitive display 110 has a corresponding desktop stack comprising desktops D3, D2, and D1,with desktop D1 being at the bottom 2300 stack position and with desktop D3 at the top stack position being capable of being displayed with window 4 (depending on the window position and size (whether maximized or minimized),

and the touch sensitive display 114 has a corresponding desktop stack has a corresponding desktop stack comprising desktops D4, D5, and D6,with desktop D6 being at the bottom 2304 stack position and with desktop D4 at the top stack position being capable of being displayed with window 5 (depending on the window position and size (whether maximized or minimized). Conceptually, the desktops can be viewed as a canvas divided, in this example, into six sections of which two can be displayed, at any one time, the touch sensitive displays 110, 114. When the device 100 is in the closed state, this conceptual model, in one configuration, persists. In this configuration, only one window and desktop stacks can be seen (that corresponding to the primary screen) but the other window and desktop stacks are virtual; that is, they are maintained in memory but cannot be seen because the secondary screen is not enabled.

[0120] Various window navigation configurations will now be discussed with reference to FIGS. 7-11. The various configurations depict examples relating to navigating among independent or hierarchically related displayed images.

[0121] Prior to discussing the examples, the concept of a root view will be described. In the examples below, a stack of first, second, and third windows 1.1, 1.1.1, and 1.1.1.1 and first and second desktops D1 and D2 are depicted. The windows are in a hierarchical tree-like sequence and are controlled by a common multi-screen application. As shown in FIG. 11, first window 1.1 is a root window, second window 1.1.1 is a node window, and third window 1.1.1.1 is a leaf window. The root window 1.1 may be a top level view of the hierarchical application window sequence such that there is no parent window corresponding to the root window. The root window may be a parent to a node window. One or more node windows may be provided that are related as parent/children. A node window may also serve as a leaf window. By leaf window, it is meant that the leaf window has no corresponding node screen for which the leaf window is a parent. As such, the leaf window does not have any children node windows, Windows 1.1.1.1 and 1.1.2 are examples of leaf windows. The root view refers to all of the windows controlled by a multi-screen application at a point in time.

[0122] In the various examples and with reference to FIG. 8, middleware 520, particularly one or more of the Multi-Display Management (MDM) class 524, a Surface Cache class 528, a Window Management class 532, an Activity Management class 536, and an Application

Management class 540, independently or collectively, detect a gesture, typically a drag, that performs a displayed image operation (step 1200) and initiates the operation (step 1204). The displayed image operation is to move a window in an active display position in an origin or first touch sensitive display 110 to a target or second touch sensitive display 114. In step 1208, the middleware determines whether the drag gesture is ongoing. If the drag gesture is ongoing, the middleware, in step 1212, determines whether there is a further window in the root view. If the answers to both of the foregoing queries is affirmative, the middleware, in step 1216, moves a next window in the origin or first touch sensitive display 110 to the target or section touch sensitive display 114 and returns to step 1208. If the drag gesture has stopped (step 1208) or if there is no further window in the root view (step 1212), the middleware logic terminates operation in step 1220.

[0123] A first example will be applied to FIGS. 7A-F, Referring again to FIGS. 7A-F, the primary and secondary touch sensitive displays 110 and 114 are in portrait display orientation and display, in active display positions, the minimized third window 1.1.1.1 and second desktop D2, respectively (FIG. 7A). A window is minimized when it is displayed by only one of the primary and secondary screens. The displayed images are part of ordered set(s) of currently or previously displayed images, with the third window 1.1.1.1 and second desktop D2 occupying active display positions and the first and second windows 1.1 and 1.1.1 and first desktop D1 occupying inactive display positions. Stated another way, the stack corresponding to the primary screen 104 has from top-to-bottom the third window 1.1.1.1, the second window 1.1.1, the first window 1.1, and the first desktop D1. The first, second, and third windows are controlled by a multi-screen application and not by separate single-screen applications. In FIG. 7B, the primary screen 104, commonly the gesture capture region 120, detects a user drag gesture 700 on the gesture capture region 120 to perform a displayed image operation. The displayed image operation is to move the third window 1.1.1.1 from the origin or first touch sensitive display 110 to the target or second touch sensitive display 114. As shown in FIG. 7B, the third window 1.1.1.1 at the top of the stack has started to move in the direction indicated to occupy an active display position (such as in a respective frame buffer or stack) or state for the secondary screen 108 and displace the second desktop to an inactive display position. In the first displayed image operation, the selected third window, but not the first and second windows 1.1 and 1.1.1, is to be moved to

the secondary screen 108. At the conclusion of the operation, the touch sensitive display 110 would contain in the active display position the second window 1.1.1 and the touch sensitive display 114 would contain the third window 1.1.1.1 (FIG. 7C). The middleware, in response to the gesture, moves the third window 1.1.1.1 to the target touch sensitive display (FIG. 7B).

[0124] After the displayed image operation is completed and the third window 1.1.1.1 is in the active position of the target touch sensitive display 114, the middleware determines that the drag gesture is ongoing and initiates sequential movement of the next window in the root view, namely the second window 1.1.1 from the origin touch sensitive display 110 to to a display position behind the third window 1.1.1.1 in the target touch sensitive display 114 (FIG. 7D). After the second window 1.1.1 is moved to the target touch sensitive display 114, the middleware determines that the drag gesture is ongoing and initiates movement of the first window 1.1 in the root view to the target touch sensitive display 114 (FIG. 7E). When the first window 1.1 has moved to the target touch sensitive display 114 to a display position behind the third window 1.1.1.1, the origin touch sensitive display 110 has the first desktop D1 in the active display position, and the target touch sensitive display 114 has the third window 1.1.1.1 in the active display position. The stack behind the third window 1.1.1.1 includes, from highest to lowest, the second window 1.1.1, the first window 1.1, and the second desktop D2 (FIG. 7F).

[0125] Referring now to FIGS. 9A-F, the primary and secondary touch sensitive displays 110 and 114 are in portrait display orientation and display minimized first window 1.1 and second desktop D2, respectively (FIG. 9A). The displayed images are part of ordered set(s) of currently or previously displayed images, with the first window 1.1 and second desktop D2 occupying active display positions and the second and third windows 1.1.1 and 1.1.1.1 and first desktop D1 occupying inactive display positions in the stack behind the first window 1.1. Stated another way, the stack corresponding to the primary screen 104 has from top-to-bottom the first window 1.1, the second window 1.1.1, the third window 1.1.1.1, and the first desktop D1. The first, second, and third windows are controlled by a multi-screen application and not by separate single-screen applications. In FIG. 9B, the primary screen 104, commonly the gesture capture region 120, detects a user gesture on the gesture capture region 120 to perform a displayed image operation. In one example, the user gesture 700 is a drag. As shown in FIG. 9B, the first window 1.1 has started to move from a position in front

of the second window 1.1.1 in the direction indicated to occupy an active display position (such as in a respective frame buffer or stack) or state for the secondary screen 108 and displace or cover the second desktop to an inactive display position. In the displayed image operation, the selected first window 1.1, but not the second and third windows 1.1.1 and 1.1.1.1, is to be moved to the secondary screen 108 as part of a window maximization transition. At the conclusion of the operation, the touch sensitive display 110 would have the second window 1.1.1 and the touch sensitive display 114 would have the first window 1.1 in their respective active display positions (FIG. 9C). The third window 1.1.1.1 would remain in an inactive display position in the stack of the primary screen 104.

[0126] At the conclusion of the displayed image operation, the middleware, however, senses that the drag gesture is ongoing. In response, the middleware causes the second window 1.1.1 to move from the origin touch sensitive display 110 to a position behind the first window 1,1 in the target touch sensitive display 114 (FIG. 9D). At the conclusion of the displayed image operation (with the second window 1.1.1 in an inactive display position behind the first window 1.1 on the target touch sensitive display 114), the middleware, however, again senses that the drag gesture is ongoing. In response, the middleware causes the third window 1.1.1 to move from the origin touch sensitive display 110 to a position behind the first window 1,1 in the target touch sensitive display 114 (FIG. 9E). When the third window completes movement to the target touch sensitive display 114, the target touch sensitive display 114 will have, as a stack, from top-to-bottom, the first window 1.1, the second window 1.1.1, the third window 1.1.1.1, and the second desktop D2. The first desktop D1 is in the active display position for the touch sensitive display 110 (FIG. 9F).

[0127] A further landscape display orientation example will be applied to FIGS. 10A-C, A stack of first, second, and third windows 1.1, 1.1.1, and 1.1.1.1 and first and second desktops D2 and D3 are depicted. The windows are in a hierarchical tree-like sequence and are controlled by a common multi-screen application.

[0128] Referring again to FIGS. 10A-C, the primary and secondary touch sensitive displays 110 and 114 are in landscape display orientation and display, in active display positions, the minimized third window 1.1.1.1 and second desktop D2, respectively (FIG. 10A). The displayed images are part of ordered set(s) of currently or previously displayed images, with the third window 1.1.1.1 and second desktop D2 occupying active display

positions and the first and second windows 1.1 and 1.1.1 and first desktop D1 occupying inactive display positions. Stated another way, the stack corresponding to the primary screen 104 has from top-to-bottom the third window 1.1.1.1, the second window 1.1.1, the first window 1.1, and the first desktop D1. The first, second, and third windows are controlled by a multi-screen application and not by separate single-screen applications. In FIG. 10B, the primary screen 104, commonly the gesture capture region 120, detects a user gesture on the gesture capture region 120 to perform a displayed image operation. In one example, the user gesture 700 is a drag. As shown in FIG. 10A, the third window 1.1.1.1 at the top of the stack has started to move in the direction indicated to occupy an active display position (such as in a respective frame buffer or stack) or state for the secondary screen 108 and displace the second desktop to an inactive display position. At the conclusion of the operation, the touch sensitive display 110 contains in the active display position the second window 1.1.1 and the touch sensitive display 114 the third window 1.1.1.1 (FIG. 10B). The middleware determines that the gesture 700 is ongoing and therefore moves the second window 1.1.1 from the origin touch sensitive display 110 to an inactive display position behind the third window 1.1.1.1 in the target touch sensitive display 114. At the conclusion of the operation, the touch sensitive display 110 contains in the active display position the first window 1.1 and the touch sensitive display 114 the third window 1.1.1.1 (view not shown). The middleware determines that the gesture 700 is ongoing and therefore moves the first window 1.1 from the origin touch sensitive display 110 to an inactive display position behind the third window 1.1.1.1 in the target touch sensitive display 114 to produce the displays of FIG. 10C.

[0129] While the window maximization transition and first displayed image operation is being performed, the primary screen 104 detects receipt of a second gesture, typically a flick gesture and typically by the gesture capture region 124. As will be appreciated, for the present disclosure to associate the first and second gestures, they commonly are received by a common gesture capture region or touch sensitive display. If the gestures are received by different ones of gesture capture region 120 and 124 or touch sensitive displays 110 and 114, associating the gestures with one another can be difficult. In isolation, a drag gesture 700 moves a desktop or window to another location on a same or different touch sensitive display 110 and 114. A flick gesture 704 can move an entire root view of the windows of a multi-screen application from one touch sensitive display to another, thereby uncovering a desktop.

[0130] The middleware 520 determines, due to the temporal and spatial association between the sensed continuation of the drag gesture and receipt of the flick gesture, that the user desires to retract the ongoing window maximization transition of the second and third windows and that the root view is to be moved to the secondary screen 108 in accordance with flick rules. As shown in FIG. 10C, the root view (the hierarchically arranged windows from a common multi-screen application), namely first, second, and third windows 1.1, 1.1.1, and 1.1.1.1, previously positioned above the first desktop D1 on the primary screen 104, have now been moved to the secondary screen 108 to uncover the first desktop D1 (FIG. 10C). Stated another way, the stack corresponding to the secondary screen 104 has from top-to-bottom the third window 1.1.1.1, the second window 1.1.1, the first window 1.1, and the second desktop D2. In the absence of the flick gesture, the drag gesture would have moved the third window 1.1.1.1 to the secondary screen 108 to reveal the second window 1.1.1 in the primary screen 104 (view not shown).

[0131] Although drag and flick gestures are depicted in FIG. 7-8, it is to be understood that different gestures may be used.

[0132] Desktop navigation is normally effected using a suitable gesture, such as a drag or flick gesture, received in the touch sensitive display. Unlike desktop navigation, gestures received by a gesture capture region 120 or 124 are used to move windows from one screen to the other (e.g., from one active or inactive position to another).

[0133] The exemplary systems and methods of this disclosure have been described in relation to communication devices. However, to avoid unnecessarily obscuring the present disclosure, the preceding description omits a number of known structures and devices. This omission is not to be construed as a limitation of the scopes of the claims. Specific details are set forth to provide an understanding of the present disclosure. It should however be appreciated that the present disclosure may be practiced in a variety of ways beyond the specific detail set forth herein.

[0134] Furthermore, while the exemplary aspects, embodiments, and/or configurations illustrated herein show the various components of the system collocated, certain components of the system can be located remotely, at distant portions of a distributed network, such as a LAN and/or the Internet, or within a dedicated system. Thus, it should be appreciated, that the components of the system can be combined in to one or more devices, such as a

communication device, or collocated on a particular node of a distributed network, such as an analog and/or digital telecommunications network, a packet-switch network, or a circuit-switched network. It will be appreciated from the preceding description, and for reasons of computational efficiency, that the components of the system can be arranged at any location within a distributed network of components without affecting the operation of the system. For example, the various components can be located in a switch such as a PBX and media server, gateway, in one or more communications devices, at one or more users' premises, or some combination thereof. Similarly, one or more functional portions of the system could be distributed between a telecommunications device(s) and an associated computing device.

[0135] Furthermore, it should be appreciated that the various links connecting the elements can be wired or wireless links, or any combination thereof, or any other known or later developed element(s) that is capable of supplying and/or communicating data to and from the connected elements. These wired or wireless links can also be secure links and may be capable of communicating encrypted information. Transmission media used as links, for example, can be any suitable carrier for electrical signals, including coaxial cables, copper wire and fiber optics, and may take the form of acoustic or light waves, such as those generated during radio-wave and infra-red data communications.

[0136] Also, while the flowcharts have been discussed and illustrated in relation to a particular sequence of events, it should be appreciated that changes, additions, and omissions to this sequence can occur without materially affecting the operation of the disclosed embodiments, configuration, and aspects.

[0137] A number of variations and modifications of the disclosure can be used. It would be possible to provide for some features of the disclosure without providing others.

[0138] For example in one alternative embodiment, the desktops are moved by gestures received in a gesture capture region 120, 124, and windows are moved by gestures received in a touch sensitive display 110, 114.

[0139] In yet another embodiment, the systems and methods of this disclosure can be implemented in conjunction with a special purpose computer, a programmed microprocessor or microcontroller and peripheral integrated circuit element(s), an ASIC or other integrated circuit, a digital signal processor, a hard-wired electronic or logic circuit such as discrete element circuit, a programmable logic device or gate array such as PLD, PLA, FPGA, PAL,

special purpose computer, any comparable means, or the like. In general, any device(s) or means capable of implementing the methodology illustrated herein can be used to implement the various aspects of this disclosure. Exemplary hardware that can be used for the disclosed embodiments, configurations and aspects includes computers, handheld devices, telephones (e.g., cellular, Internet enabled, digital, analog, hybrids, and others), and other hardware known in the art. Some of these devices include processors (e.g., a single or multiple microprocessors), memory, nonvolatile storage, input devices, and output devices. Furthermore, alternative software implementations including, but not limited to, distributed processing or component/object distributed processing, parallel processing, or virtual machine processing can also be constructed to implement the methods described herein.

[0140] In yet another embodiment, the disclosed methods may be readily implemented in conjunction with software using object or object-oriented software development environments that provide portable source code that can be used on a variety of computer or workstation platforms. Alternatively, the disclosed system may be implemented partially or fully in hardware using standard logic circuits or VLSI design. Whether software or hardware is used to implement the systems in accordance with this disclosure is dependent on the speed and/or efficiency requirements of the system, the particular function, and the particular software or hardware systems or microprocessor or microcomputer systems being utilized.

[0141] In yet another embodiment, the disclosed methods may be partially implemented in software that can be stored on a storage medium, executed on programmed general-purpose computer with the cooperation of a controller and memory, a special purpose computer, a microprocessor, or the like. In these instances, the systems and methods of this disclosure can be implemented as program embedded on personal computer such as an applet, JAVA® or CGI script, as a resource residing on a server or computer workstation, as a routine embedded in a dedicated measurement system, system component, or the like. The system can also be implemented by physically incorporating the system and/or method into a software and/or hardware system.

[0142] Although the present disclosure describes components and functions implemented in the aspects, embodiments, and/or configurations with reference to particular standards and protocols, the aspects, embodiments, and/or configurations are not limited to such standards

and protocols. Other similar standards and protocols not mentioned herein are in existence and are considered to be included in the present disclosure. Moreover, the standards and protocols mentioned herein and other similar standards and protocols not mentioned herein are periodically superseded by faster or more effective equivalents having essentially the same functions. Such replacement standards and protocols having the same functions are considered equivalents included in the present disclosure.

[0143] The present disclosure, in various aspects, embodiments, and/or configurations, includes components, methods, processes, systems and/or apparatus substantially as depicted and described herein, including various aspects, embodiments, configurations embodiments, subcombinations, and/or subsets thereof. Those of skill in the art will understand how to make and use the disclosed aspects, embodiments, and/or configurations afrer understanding the present disclosure. The present disclosure, in various aspects, embodiments, and/or configurations, includes providing devices and processes in the absence of items not depicted and/or described herein or in various aspects, embodiments, and/or configurations hereof, including in the absence of such items as may have been used in previous devices or processes, e.g., for improving performance, achieving ease and\or reducing cost of implementation.

[0144] The foregoing discussion has been presented for purposes of illustration and description. The foregoing is not intended to limit the disclosure to the form or forms disclosed herein. In the foregoing Detailed Description for example, various features of the disclosure are grouped together in one or more aspects, embodiments, and/or configurations for the purpose of streamlining the disclosure. The features of the aspects, embodiments, and/or configurations of the disclosure may be combined in alternate aspects, embodiments, and/or configurations other than those discussed above. This method of disclosure is not to be interpreted as reflecting an intention that the claims require more features than are expressly recited in each claim. Rather, as the following claims reflect, inventive aspects lie in less than all features of a single foregoing disclosed aspect, embodiment, and/or configuration. Thus, the following claims are hereby incorporated into this Detailed Description, with each claim standing on its own as a separate preferred embodiment of the disclosure.

[0145] Moreover, though the description has included description of one or more aspects, embodiments, and/or configurations and certain variations and modifications, other variations, combinations, and modifications are within the scope of the disclosure, e.g., as may be within the skill and knowledge of those in the art, after understanding the present disclosure. It is intended to obtain rights which include alternative aspects, embodiments, and/or configurations to the extent permitted, including alternate, interchangeable and/or equivalent structures, functions, ranges or steps to those claimed, whether or not such alternate, interchangeable and/or equivalent structures, functions, ranges or steps are disclosed herein, and without intending to publicly dedicate any patentable subject matter.

What is claimed is:

1. A method, comprising:

receiving, by at least one of a gesture capture region and a touch sensitive display, a gesture; and

in response, moving, by processor-executable middleware, a first displayed image on a first touch sensitive display of a communication device to a second touch sensitive display of the communication device;

determining, after the first displayed image has moved successfully to the second touch sensitive display and by processor-executable middleware,, that the gesture is continuing; and

in response, moving, by processor-executable middleware, a second displayed image on a first touch sensitive display of a communication device to the second touch sensitive display of the communication device wherein the first and second displayed images are at least one of a desktop and window.

2. The method of claim 1, wherein the first and second displayed images are windows and wherein the first gesture is a drag.

3. The method of claim 1, wherein the first and second displayed images are controlled by a common multi-screen application and wherein, in response to the gesture, both the first and second displayed images are moved ftom the first touch sensitive display to the second touch sensitive display.

4. The method of claim 1, wherein stack positions of the first and second displayed images on the first touch sensitive display before the gesture are the same as stack positions of the first and second displayed images after movement to the second touch sensitive display.

5. The method of claim 1, wherein the first and second gestures are received by a common gesture capture region and wherein the gesture capture region is unable to display any displayed image.

6. The method of claim 1, wherein the first and second displayed images are windows controlled by a common multi-screen application and wherein a relative ordering of the windows is the same before and after movement.

7. A non-transient computer readable medium comprising processor executable instructions operable to perform the steps of claim 1.

8. A method, comprising:

receiving, by a gesture capture region, a drag gesture;

in response, moving, by processor-executable middleware, a first window ftom a first touch sensitive display to a second touch sensitive display;

determining, by the processor-executable middleware, that the drag gesture is continuous; and

in response, moving, by the processor-executable middleware, a second window ftom the first touch sensitive display to the second touch sensitive display.

9. The method of claim 8, wherein the first and second windows are controlled by a common multi-screen application.

10. The method of claim 8, wherein the first and second windows are hierarchically related to one another.

11. The method of claim 8, wherein the drag gestures is received by a gesture capture region and wherein the gesture capture region is unable to display any displayed image and wherein the first and second displayed images are windows controlled by a common multi-screen application.

12. A non-transient computer readable medium comprising processor executable instructions operable to perform the steps of claim 8.

13. A dual display communication device, comprising:

a gesture capture region to receive a flick gesture;

a first touch sensitive display to receive a flick gesture and display displayed images, wherein the displayed images are at least one of a desktop and window of an application;

a second touch sensitive display to receive a flick gesture and display displayed images; and

middleware operable to perform at least one of the following operations:

receive, by at least one of the gesture capture region and a touch sensitive display, a gesture; and

in response, move a first displayed image on the first touch sensitive display to the second touch sensitive display;

determine, after the first displayed image has moved successfully to the second touch sensitive display, that the gesture is continuing; and

in response, move a second displayed image on the first touch sensitive display to the second touch sensitive display, wherein the first and second displayed images are at least one of a desktop and window.

14. The device of claim 13 wherein the first and second displayed images are windows and wherein the first gesture is a drag.

15. The device of claim 13, wherein the first and second displayed images are controlled by a common multi-screen application and wherein, in response to the gesture, both the first and second displayed images are moved ftom the first touch sensitive display to the second touch sensitive display.

16. The device of claim 13, wherein stack positions of the first and second displayed images on the first touch sensitive display before the gesture are the same as stack positions of the first and second displayed images after movement to the second touch sensitive display.

17. The device of claim 13, wherein the first and second gestures are received by a common gesture capture region and wherein the gesture capture region is unable to display any displayed image.

18. The device of claim 13, wherein the first and second displayed images are windows controlled by a common multi-screen application and wherein a relative ordering of the windows is the same before and after movement.

ABSTRACT

[0146] The disclosed method and device are directed to navigation, by a dual display communication device, through display objects.

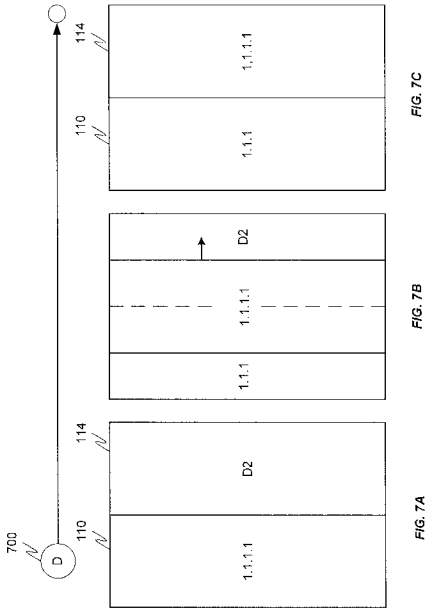


FIG. 7A

FIG. 7B

FIG. 7C

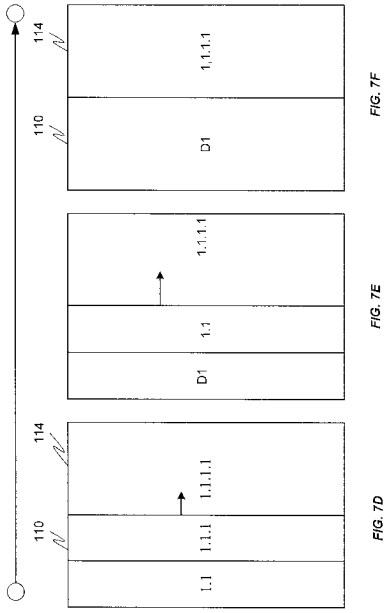


FIG. 7D

FIG. 7E

FIG. 7F

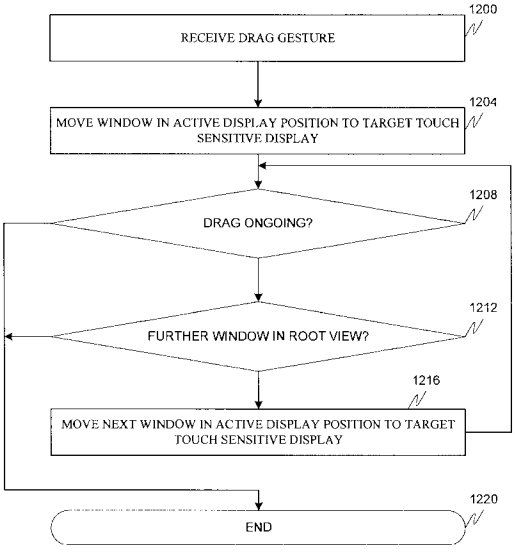


FIG. 8

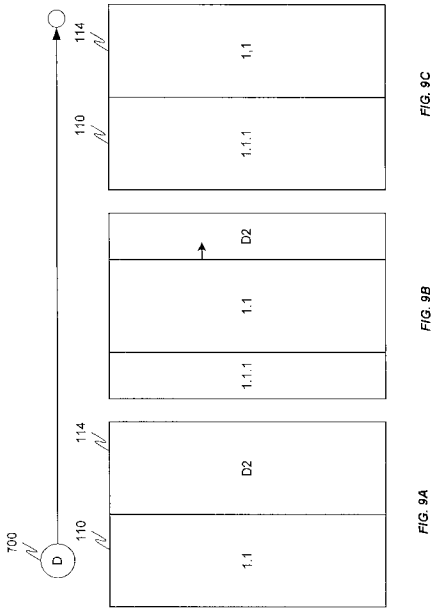


FIG. 9A

FIG. 9B

FIG. 9C

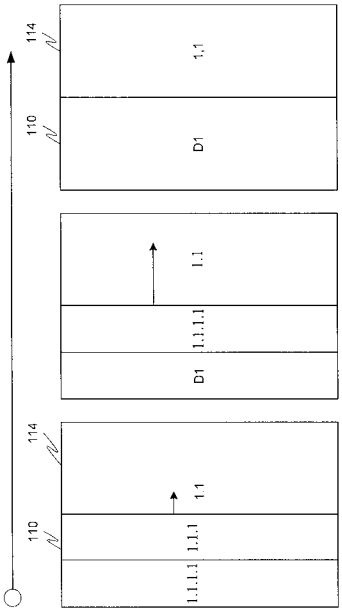


FIG. 9F FIG. 9E FIG. 9D

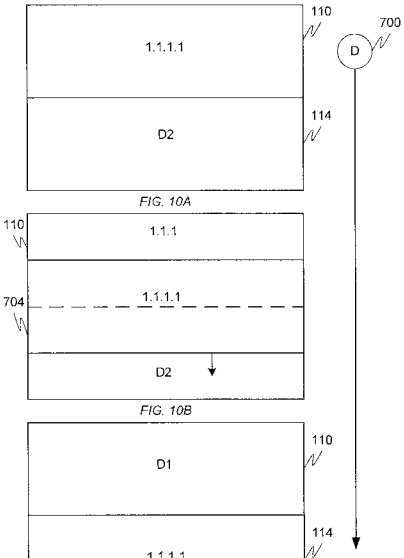


FIG. 10A

FIG. 10B

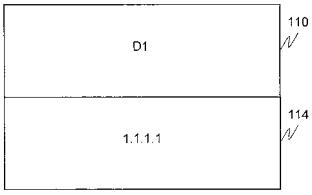


FIG. 10C

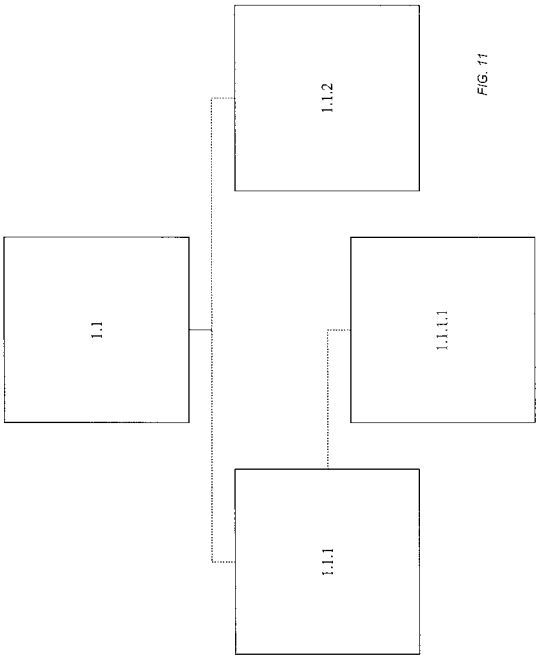


FIG. 11

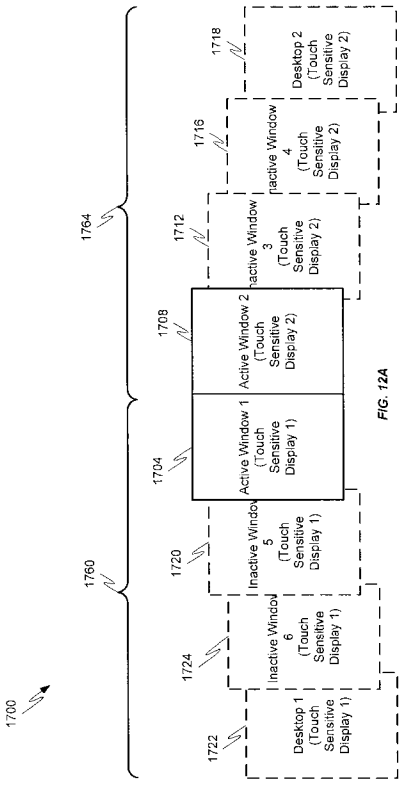


FIG. 12A

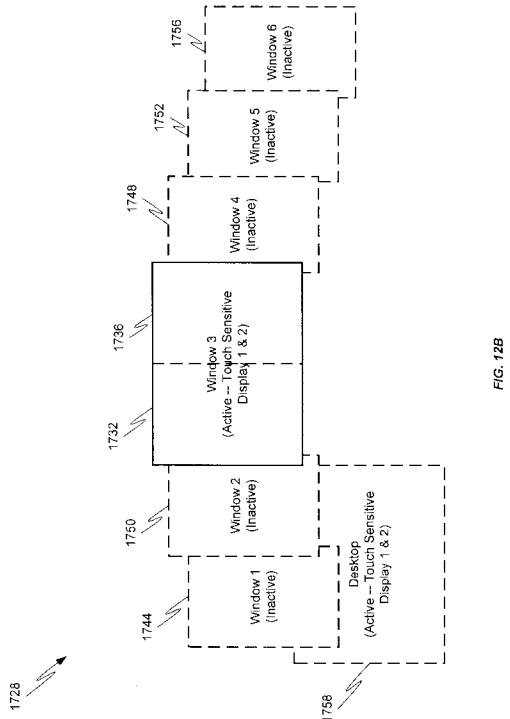


FIG. 12B

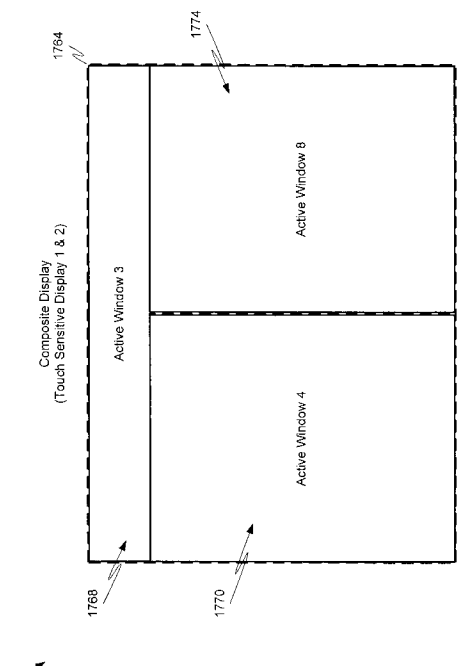


FIG. 12C

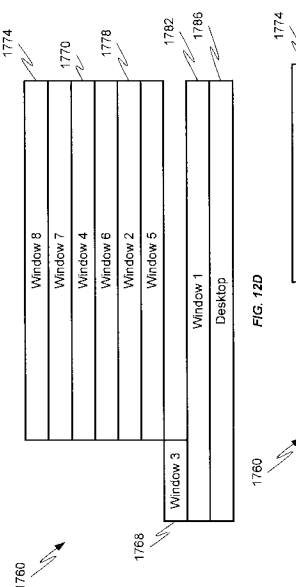


FIG. 12D

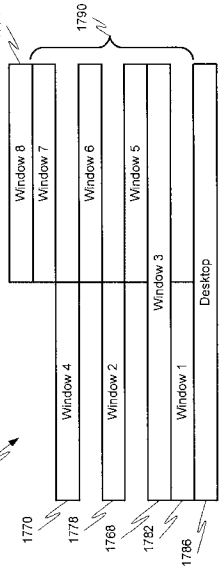


FIG. 12E

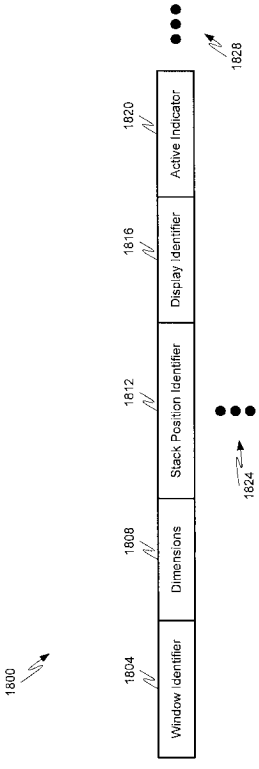


FIG. 13

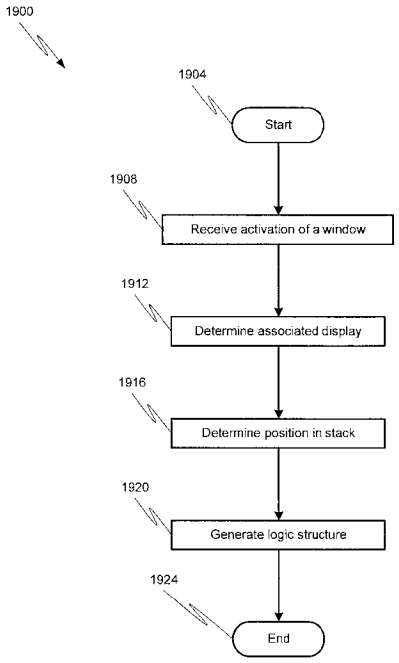

1900
1904
Start
1908
Receive activation of a window
1912
Determine associated display
1916
Determine position in stack
1920
Generate logic structure
1924
End

FIG. 14

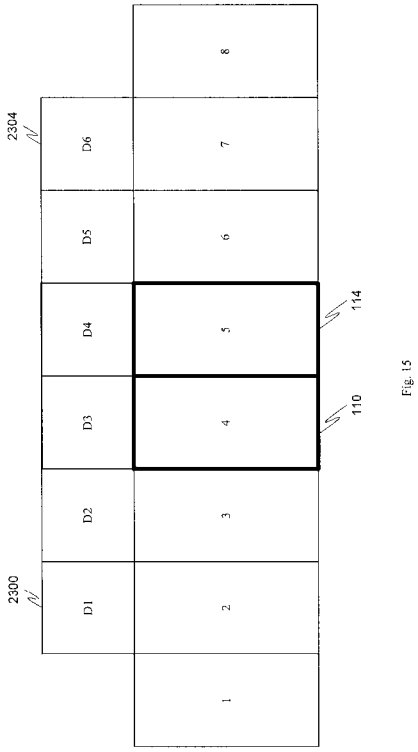

2304
D6
D5
D4
D3
D2
D1
2300
8
7
6
5
4
3
2
1
114
110

Fig. 15

DRAG/FLICK GESTURES IN USER INTERFACE

[0100] The device 100 manages desktops and/or windows with at least one window stack 1700, 1728, as shown in Figs. 12A and 12B. A window stack 1700, 1728 is a logical arrangement of active and/or inactive windows for a multi-screen device. For example, the window stack 1700, 1728 may be logically similar to a deck of cards, where one or more windows or desktops are arranged in order, as shown in Figs. 12A and 12B. An active window is a window that is currently being displayed on at least one of the touch sensitive displays 110, 114. For example, windows 104 and 108 are active windows and are displayed on touch sensitive displays 110 and 114. An inactive window is a window that was opened and displayed but is now "behind" an active window and not being displayed. In embodiments, an inactive window may be for an application that is suspended, and thus, the window is not displaying active content. For example, windows 1712, 1716, 1720, and 1724 are inactive windows.

[0101] A window stack 1700, 1728 may have various arrangements or organizational structures. In the embodiment shown in Fig. 12A, the device 100 includes a first stack 1760 associated with a first touch sensitive display 110 and a second stack associated with a second touch sensitive display 114. Thus, each touch sensitive display 110, 114 can have an associated window stack 1760, 1764. These two window stacks 1760, 1764 may have different numbers of windows arranged in the respective stacks 1760, 1764. Further, the two window stacks 1760, 1764 can also be identified differently and managed separately. Thus, the first window stack 1760 can be arranged in order from a first window 1704 to a next window 1720 to a last window 1724 and finally to a desktop 1722, which, in embodiments, is at the "bottom" of the window stack 1760. In embodiments, the desktop 1722 is not always at the "bottom" as application windows can be arranged in the window stack below the desktop 1722, and the desktop 1722 can be brought to the "top" of a stack over other windows during a desktop reveal. Likewise, the second stack 1764 can be arranged from a first window 1708 to a next window 1712 to a last window 1716, and finally to a desktop 1718, which, in embodiments, is a single desktop area, with desktop 1722, under all the windows in both window stack 1760 and window stack 1764. A logical data structure for managing the two window stacks 1760, 1764 may be as described in conjunction with Fig. 13.

[0102] Another arrangement for a window stack 1728 is shown in Fig. 12B. In this embodiment, there is a single window stack 1728 for both touch sensitive displays 110, 114.

Thus, the window stack 1728 is arranged from a desktop 1758 to a first window 1744 to a last window 1756. A window can be arranged in a position among all windows without an association to a specific touch sensitive display 110, 114. In this embodiment, a window is in the order of windows. Further, at least one window is identified as being active. For example, a single window may be rendered in two portions 1732 and 1736 that are displayed on the first touch sensitive screen 110 and the second touch sensitive screen 114. The single window may only occupy a single position in the window stack 1728 although it is displayed on both displays 110, 114.

[0103] Yet another arrangement of a window stack 1760 is shown in Figs. 12C through 12E. The window stack 1760 is shown in three "elevation" views. In Fig. 12C, the top of the window stack 1760 is shown. Two sides of the window stack 1760 are shown in Figs. 12D and 12E. In this embodiment, the window stack 1760 resembles a stack of bricks. The windows are stacked on each other. Looking from the top of the window stack 1760 in Fig, 12C, only the top most windows in the window stack 1760 are seen in different portions of the composite display 1764. The composite display 1764 represents a logical model for the entire display area of the device 100, which can include touch sensitive display 110 and touch sensitive display 114. A desktop 1786 or a window can occupy part or all of the composite display 1764.

[0104] In the embodiment shown, the desktop 1786 is the lowest display or "brick" in the window stack 1760. Thereupon, window 1 1782, window 2 1782, window 3 1768, and window 4 1770 are layered. Window 1 1782, window 3 1768, window 2 1782, and window 4 1770 only occupy a portion of the composite display 1764. Thus, another part of the stack 1760 includes window 8 1774 and windows 5 through 7 shown in section 1790. Only the top window in any portion of the composite display 1764 is actually rendered and displayed. Thus, as shown in the top view in Fig. 12C, window 4 1770, window 8 1774, and window 3 1768 are displayed as being at the top of the display in different portions of the window stack 1760. A window can be dimensioned to occupy only a portion of the composite display 1760 to "reveal" windows lower in the window stack 1760. For example, window 3 1768 is lower in the stack than both window 4 1770 and window 8 1774 but is still displayed. A logical data structure to manage the window stack can be as described in conjunction with Fig. 13.

[0105] When a new window is opened, the newly activated window is generally positioned at the top of the stack. However, where and how the window is positioned within the stack can be

a function of the orientation of the device 100, the context of what programs, functions, software, etc. are being executed on the device 100, how the stack is positioned when the new window is opened, etc. To insert the window in the stack, the position in the stack for the window is determined and the touch sensitive display 110, 114 to which the window is associated may also be determined. With this information, a logical data structure for the window can be created and stored. When user interface or other events or tasks change the arrangement of windows, the window stack(s) can be changed to reflect the change in arrangement. It should be noted that these same concepts described above can be used to manage the one or more desktops for the device 100.

[0106] A logical data structure 1800 for managing the arrangement of windows or desktops in a window stack is shown in Fig. 13. The logical data structure 1800 can be any data structure used to store data whether an object, record, file, etc. The logical data structure 1800 can be stored in any type of database or data storage system, regardless of protocol or standard. In embodiments, the logical data structure 1800 includes one or more portions, fields, attributes, etc. that store data in a logical arrangement that allows for easy storage and retrieval of the information. Hereinafrer, these one or more portions, fields, attributes, etc. shall be described simply as fields. The fields can store data for a window identifier 1804, dimensions 1808, a stack position identifier 1812, a display identifier 1816, and/or an active indicator 1820. Each window in a window stack can have an associated logical data structure 1800. While only a single logical data structure 1800 is shown in Fig. 13, there may be more or fewer logical data structures 1800 used with a window stack (based on the number of windows or desktops in the stack), as represented by ellipses 1824. Further, there may be more or fewer fields than those shown in Fig. 13, as represented by ellipses 1828.

[0107] A window identifier 1804 can include any identifier (ID) that uniquely identifies the associated window in relation to other windows in the window stack. The window identifier 1804 can be a globally unique identifier (GUID), a numeric ID, an alphanumeric ID, or other type of identifier. In embodiments, the window identifier 1804 can be one, two, or any number of digits based on the number of windows that can be opened. In alternative embodiments, the size of the window identifier 1804 may change based on the number of windows opened. While the window is open, the window identifier 1804 may be static and remain unchanged.

[0108] Dimensions 1808 can include dimensions for a window in the composite display 1760. For example, the dimensions 1808 can include coordinates for two or more corners of the window or may include one coordinate and dimensions for the width and height of the window. These dimensions 1808 can delineate what portion of the composite display 1760 the window may occupy, which may the entire composite display 1760 or only part of composite display 1760. For example, window 4 1770 may have dimensions 1880 that indicate that the window 1770 will occupy only part of the display area for composite display 1760, as shown in Figs. 12C through 12E. As windows are moved or inserted in the window stack, the dimensions 1808 may change.

[0109] A stack position identifier 1812 can be any identifier that can identify the position in the stack for the window or may be inferred from the window's control record within a data structure, such as a list or a stack. The stack position identifier 1812 can be a GUID, a numeric ID, an alphanumeric ID, or other type of identifier. Each window or desktop can include a stack position identifier 1812. For example, as shown in Fig. 12A, window 1 1704 in stack 1 1760 can have a stack position identifier 1812 of 1 identifying that window 1704 is the first window in the stack 1760 and the active window. Similarly, window 6 1724 can have a stack position identifier 1812 of 3 representing that window 1724 is the third window in the stack 1760. Window 2 1708 can also have a stack position identifier 1812 of 1 representing that window 1708 is the first window in the second stack 1764. As shown in Fig. 12B, window 1 1744 can have a stack position identifier 1812 of 1, window 3, rendered in portions 1732 and 1736, can have a stack position identifier 1812 of 3, and window 6 1756 can have a stack position identifier 1812 of 6. Thus, depending on the type of stack, the stack position identifier 1812 can represent a window's location in the stack.

[0110] A display identifier 1816 can identify that the window or desktop is associated with a particular display, such as the first display 110 or the second display 114, or the composite display 1760 composed of both displays. While this display identifier 1816 may not be needed for a multi-stack system, as shown in Fig. 12A, the display identifier 1816 can indicate whether a window in the serial stack of Fig. 12B is displayed on a particular display. Thus, window 3 may have two portions 1732 and 1736 in Fig. 12B. The first portion 1732 may have a display identifier 1816 for the first display while the second portion 1736 may have a display identifier 1816 for the second display 114. However, in alternative embodiments, the window may have

two display identifier 1816 that represent that the window is displayed on both of the displays 110, 114, or a display identifier 1816 identifying the composite display. In another alternate embodiment, the window may have a single display identifier 1816 to represent that the window is displayed on both of the displays 110, 114.

[0111] Similar to the display identifier 1816, an active indicator 1820 may not be needed with the dual stack system of Fig. 12A, as the window in stack position 1 is active and displayed. In the system of Fig. 12B, the active indicator 1820 can indicate which window(s) in the stack is being displayed. Thus, window 3 may have two portions 1732 and 1736 in Fig. 12. The first portion 1732 may have an active indicator 1820 while the second portion 1736 may also have an active indicator 1820. However, in alternative embodiments, window 3 may have a single active indicator 1820. The active indicator 1820 can be a simple flag or bit that represents that the window is active or displayed.

[0112] An embodiment of a method 1900 for creating a window stack is shown in Fig. 14. While a general order for the steps of the method 1900 is shown in Fig. 14. Generally, the method 1900 starts with a start operation 1904 and ends with an end operation 1928. The method 1900 can include more or fewer steps or can arrange the order of the steps differently than those shown in Fig. 14. The method 1900 can be executed as a set of computer-executable instructions executed by a computer system and encoded or stored on a computer readable medium. Hereinafrer, the method 1900 shall be explained with reference to the systems, components, modules, software, data structures, user interfaces, etc. described in conjunction with Figs. 1-13.

[0113] A multi-screen device 100 can receive activation of a window, in step 1908. In embodiments, the multi-screen device 100 can receive activation of a window by receiving an input from the touch sensitive display 110 or 114, the configurable area 112 or 116, a gesture capture region 120 or 124, or some other hardware sensor operable to receive user interface inputs. The processor may execute the Task Management Module 540 may receive the input. The Task Management Module 540 can interpret the input as requesting an application task to be executed that will open a window in the window stack.

[0114] In embodiments, the Task Management Module 540 places the user interface interaction in the task stack 552 to be acted upon by the Display Configuration Module 568 of the Multi-Display Management Module 524. Further, the Task Management Module 540 waits

for information from the Multi-Display Management Module 524 to send instructions to the Window Management Module 532 to create the window in the window stack.

[0115] The Multi-Display Management Module 524, upon receiving instruction from the Task Management Module 540, determines to which touch portion of the composite display 1760, the newly activated window should be associated, in step 1912. For example, window 4 1770 is associated with the a portion of the composite display 1764 In embodiments, the device state module 574 of the Multi-Display Management Module 524 may determine how the device is oriented or in what state the device is in, e.g., open, closed, portrait, etc. Further, the preferences module 572 and/or requirements module 580 may determine how the window is to be displayed. The gesture module 576 may determine the user's intentions about how the window is to be opened based on the type of gesture and the location of where the gesture is made.

[0116] The Display Configuration Module 568 may use the input from these modules and evaluate the current window stack 1760 to determine the best place and the best dimensions, based on a visibility algorithm, to open the window. Thus, the Display Configuration Module 568 determines the best place to put the window at the top of the window stack 1760, in step 1916. The visibility algorithm, in embodiments, determines for all portions of the composite display, which windows are at the top of the stack. For example, the visibility algorithm determines that window 3 1768, window 4 1770, and window 8 1774 are at the top of the stack 1760 as viewed in Figs. 12C through 12E. Upon determining where to open the window, the Display Configuration Module 568 can assign a display identifier 816 and possibly dimensions 808 to the window. The display identifier 816 and dimensions 808 can then be sent back to the Task Management Module 540. The Task Management Module 540 may then assign the window a stack position identifier 812 indicating the windows position at the top of the window stack.

[0117] In embodiments, the Task Management Module 540 sends the window stack information and instructions to render the window to the Window Management Module 532. The Window Management Module 532 and the Task Management Module 540 can create the logical data structure 800, in step 1924. Both the Task Management Module 540 and the Window Management Module 532 may create and manage copies of the window stack. These copies of the window stack can be synchronized or kept similar through communications between the Window Management Module 532 and the Task Management Module 540. Thus,

the Window Management Module 532 and the Task Management Module 540, based on the information determined by the Multi-Display Management Module 524, can assign dimensions 808, a stack position identifier 812 (e.g., window 1 1782, window 4 1770, etc.), a display identifier 816 (e.g., touch sensitive display 1 110, touch sensitive display 2 114, composite display identifier, etc,), and an active indicator 820, which is generally always set when the window is at the "top" of the stack. The logical data structure 800 may then be stored by both the Window Management Module 532 and the Task Management Module 540. Further, the Window Management Module 532 and the Task Management Module 540may thereinafrer manage the window stack and the logical data structure(s) 800.

[0118] FIG. 15 depicts a further window stacking configuration. A plurality of windows 1, 2, 3, 4, 5, 6, 7, and 8, whether from the same or different multi-screen or single-screen applications, are depicted. The touch sensitive display 110 currently has window 4 in the active display position while touch sensitive display 114 currently has window 5 in the active display position. The stack for touch sensitive display 110, from top to bottom, has window 4 in the active display position and windows 3, 2, and 1 positioned behind it. The stack for touch sensitive display 114, from top to bottom, has window 5 in the active display position and windows 6, 7, and 8 positioned behind it.

[0119] The desktops D1, D2, D3, D4, D5, and D6 are positioned behind the window stacks. The desktops can be viewed as a desktop stack different from the window stack. Viewed in this manner, the touch sensitive display 110 has a corresponding desktop stack comprising desktops D3, D2, and D1,with desktop D1 being at the bottom 2300 stack position and with desktop D3 at the top stack position being capable of being displayed with window 4 (depending on the window position and size (whether maximized or minimized), and the touch sensitive display 114 has a corresponding desktop stack has a corresponding desktop stack comprising desktops D4, D5, and D6,with desktop D6 being at the bottom 2304 stack position and with desktop D4 at the top stack position being capable of being displayed with window 5 (depending on the window position and size (whether maximized or minimized). Conceptually, the desktops can be viewed as a canvas divided, in this example, into six sections of which two can be displayed, at any one time, the touch sensitive displays 110, 114. When the device 100 is in the closed state, this conceptual model, in one configuration, persists. In this configuration, only one window and desktop stacks can be seen (that corresponding to the primary screen) but the other

window and desktop stacks are virtual; that is, they are maintained in memory but cannot be seen because the secondary screen is not enabled.

[0120] Various window navigation configurations will now be discussed with reference to FIGS. 7-10. The various configurations depict examples relating to navigating among independent or hierarchically related displayed images. In the various examples, middleware 520, particularly one or more of the Multi-Display Management (MDM) class 524, a Surface Cache class 528, a Window Management class 532, an Activity Management class 536, and an Application Management class 540, independently or collectively, detect a first gesture, typically a drag that performs a first displayed image operation (step 804) and initiates the operation (step 808). Before the operation is completed, middleware 520 detects a different second gesture, such as a flick. In response, the middleware 520 applies rules to determine whether the second gesture takes precedence over the first gesture (decision diamond 816). When the second gesture takes precedence, the first displayed image operation is interrupted and suspended. In step 820, the second displayed image operation is performed. When the second gesture does not take precedence over the first gesture and after the first displayed image operation is completed or afrer completion of step 820, the logic terminates in step 824.

[0121] A first example will be applied to Fig. 7. A stack of minimized first, second, and third windows 1.1, 1.1.1, and 1.1.1.1 and first and second desktops D1 and D2 are depicted. The windows are minimized in that they occupy only one of the primary and secondary screens 104 and 108. The windows are in a hierarchical tree-like sequence and are controlled by a common multi-screen application. As shown in Fig. 11, first window 1.1 is a root window, second window 1.1.1 is a node window, and third window 1.1.1.1 is a leaf window. The root window 1.1 may be a top level view of the hierarchical application window sequence such that there is no parent window corresponding to the root window. The root window may be a parent to a node window. One or more node windows may be provided that are related as parent/children. A node window may also serve as a leaf window. By leaf window, it is meant that the leaf window has no corresponding node screen for which the leaf window is a parent. As such, the leaf window does not have any children node windows, Windows 1.1.1.1 and 1.1.2 are examples of leaf windows. The root view refers to all of the windows controlled by a multi-screen application at a point in time.

[0122] Referring again to FIGS. 7A-C, the primary and secondary touch sensitive displays 110 and 114 are in portrait display orientation and display, in active display positions, the third window 1.1.1.1 and second desktop D2, respectively (Fig. 7A). The displayed images are part of ordered set(s) of currently or previously displayed images, with the third window 1.1.1.1 and second desktop D2 occupying active display positions and the first and second windows 1.1 and 1.1.1 and first desktop D1 occupying inactive display positions. Stated another way, the stack corresponding to the primary screen 104 has from top-to-bottom the third window 1.1.1.1, the second window 1.1.1, the first window 1.1, and the first desktop D1. The first, second, and third windows are controlled by a multi-screen application and not by separate single-screen applications. In Fig. 7B, the primary screen 104, commonly the gesture capture region 120, detects a user gesture on the gesture capture region 120 to perform a first displayed image operation. In one example, the user gesture 700 is a drag. As shown in Fig. 7A, the third window 1.1.1.1 at the top of the stack has started to move in the direction indicated to occupy an active display position (such as in a respective frame buffer or stack) or state for the secondary screen 108 and displace the second desktop to an inactive display position. In the first displayed image operation, the selected third window, but not the first and second windows 1.1 and 1.1.1, is to be moved to the secondary screen 108 as part of a window maximization transition (so that the third window occupies simultaneously the primary and secondary screens). At the conclusion of the operation, the touch sensitive display 110 would contain in the active display position for both touch sensitive displays 110 and 114 the third window 1.1.1.1.

[0123] While the window maximization transition and first displayed image operation is being performed, the primary screen 104 detects receipt of a second gesture, typically a flick gesture and typically by a gesture capture region 120 or 124. As will be appreciated, for the present disclosure to associate the first and second gestures, they commonly are received by a common gesture capture region or touch sensitive display. If the gestures are received by different ones of gesture capture region 120 and 124 or touch sensitive displays 110 and 114, associating the gestures with one another can be difficult. In isolation, a drag gesture 700 moves a desktop or window to another location on a same or different touch sensitive display 110 and 114 or maximizes the window. A flick gesture 704 can move an entire root view of the windows of a multi-screen application from one touch sensitive display to another, thereby uncovering a desktop.

[0124] The middleware 520 determines, due to the temporal and spatial association between the sensed continuation of the drag gesture and receipt of the flick gesture, that the user desires to retract the ongoing window maximization transition of the second and third windows and that the root view is to be moved to the secondary screen 108 in accordance with flick rules. As shown in Fig. 7C, the root view (the hierarchically arranged windows from a common multi-screen application), namely first, second, and third windows 1.1, 1.1.1, and 1.1.1.1, previously positioned above the first desktop D1 on the primary screen 104, have now been moved to the secondary screen 108 to uncover the first desktop D1 (Fig. 7C). Stated another way, the stack corresponding to the secondary screen 104 has from top-to-bottom the third window 1.1.1.1, the second window 1.1.1, the first window 1.1, and the second desktop D2. In the absence of the flick gesture, the drag gesture would have moved the third window 1.1.1.1 to cover the primary and secondary screens 104, 108 or to move the third window to the secondary screen 108 to reveal the second window 1.1.1 in the primary screen 104 (view not shown).

[0125] Referring now to FIGS. 9A-C, the primary and secondary touch sensitive displays 110 and 114 are in portrait display orientation and display first window 1.1 and second desktop D2, respectively (Fig. 9A). The displayed images are part of ordered set(s) of currently or previously displayed images, with the first window 1.1 and second desktop D2 occupying active display positions and the second and third windows 1.1.1 and 1.1.1.1 and first desktop D1 occupying inactive display positions. Stated another way, the stack corresponding to the primary screen 104 has from top-to-bottom the first window 1.1, the second window 1.1.1, the third window 1.1.1.1, and the first desktop D1. The first, second, and third windows are controlled by a multi-screen application and not by separate single-screen applications. In Fig. 9B, the primary screen 104, commonly the gesture capture region 120 or 124, detects a user gesture on the gesture capture region 120 or 124 to perform a first displayed image operation. In one example, the user gesture 700 is a drag. As shown in Fig. 9A, the second window 1.1.1 has started to move outward from behind the first window 1.1 in the direction indicated to occupy an active display position (such as in a respective frame buffer or stack) or state for the secondary screen 108 and displace or cover the second desktop to an inactive display position. In the first displayed image operation, the selected second window 1.1.1, but not the first and third windows 1.1 and 1.1.1.1, is to be moved to the secondary screen 108 as part of a window maximization transition. At the conclusion of the operation, the touch sensitive display 110 would contain the first window 1.1

and the touch sensitive display 114 would contain the second window 1.1.1 or the second window would occupy simultaneously both touch sensitive displays 110, 114. The third window 1.1.1.1 would remain in the stack of the primary screen 104.

[0126] While the maximization transition and first displayed image operation is being performed, the primary screen 104 detects receipt of a second gesture, typically a flick gesture and typically by the gesture capture region 124. As will be appreciated, for the present disclosure to associate the first and second gestures, they commonly are received by a common gesture capture region or touch sensitive display. If the gestures are received by different ones of gesture capture region 120 and 124, or touch sensitive displays 110 and 114, associating the gestures with one another can be difficult.

[0127] The middleware 520 determines, due to the temporal and spatial association between the sensed continuation of the drag gesture and receipt of the flick gesture, that the user desires to retract the ongoing window maximization transition of the second window 1.1.1 and that the root view (the hierarchically arranged windows from a common multi-screen application) is to be moved to the secondary screen 108 in accordance with flick rules. As shown in Fig. 9C, the root view, namely first, second, and third windows 1.1, 1.1.1, and 1.1.1.1, previously positioned above the first desktop D1 on the primary screen 104, have now been moved to the secondary screen 108 to uncover the first desktop D1 (Fig. 9C). In the absence of the flick gesture, the drag gesture would have moved the second window 1.1.1 to the secondary screen 108 to reveal the first window 1.1 in the primary screen 104 (view not shown) or maximized the second window 1.1.1.

[0128] A further landscape display orientation example will be applied to FIGS. 10A-C. A stack of first, second, and third windows 1.1, 1.1.1, and 1.1.1.1 and first and second desktops D2 and D3 are depicted. The windows are in a hierarchical tree-like sequence and are controlled by a common multi-screen application.

[0129] Referring again to FIGS. 10A-C, the primary and secondary touch sensitive displays 110 and 114 are in landscape display orientation and display, in active display positions, the third window 1.1.1.1 and second desktop D2, respectively (Fig. 10A). The displayed images are part of ordered set(s) of currently or previously displayed images, with the third window 1.1.1.1 and second desktop D2 occupying active display positions and the first and second windows 1.1 and 1.1.1 and first desktop D1 occupying inactive display positions. Stated another way, the stack

corresponding to the primary screen 104 has from top-to-bottom the third window 1.1.1.1, the second window 1.1.1, the first window 1.1, and the first desktop D1. The first, second, and third windows are controlled by a multi-screen application and not by separate single-screen applications. In Fig. 10B, the primary screen 104, commonly the gesture capture region 120, detects a user gesture on the gesture capture region 120 to perform a first displayed image operation. In one example, the user gesture 700 is a drag. As shown in Fig. 10A, the third window 1.1.1.1 at the top of the stack has started to move in the direction indicated to occupy an active display position (such as in a respective frame buffer or stack) or state for the secondary screen 108 and displace the second desktop to an inactive display position. In the first displayed image operation, the selected third window, but not the first and second windows 1.1 and 1.1.1, is to be moved to the secondary screen 108 as part of a window maximization transition. At the conclusion of the operation, the touch sensitive display 110 would contain in the active display position the second window 1.1.1 and the touch sensitive display 114 would contain the third window 1.1.1.1.

[0130] While the window maximization transition and first displayed image operation is being performed, the primary screen 104 detects receipt of a second gesture, typically a flick gesture and typically by the gesture capture region 124. As will be appreciated, for the present disclosure to associate the first and second gestures, they commonly are received by a common gesture capture region or touch sensitive display. If the gestures are received by different ones of gesture capture region 120 and 124, or touch sensitive displays 110 and 114, associating the gestures with one another can be difficult. In isolation, a drag gesture 700 moves a desktop or window to another location on a same or different touch sensitive display 110 and 114. A flick gesture 704 can move an entire root view of the windows of a multi-screen application from one touch sensitive display to another, thereby uncovering a desktop.

[0131] The middleware 520 determines, due to the temporal and spatial association between the sensed continuation of the drag gesture and receipt of the flick gesture, that the user desires to retract the ongoing window maximization transition of the second and third windows and that the root view is to be moved to the secondary screen 108 in accordance with flick rules. As shown in Fig. 10C, the root view (the hierarchically arranged windows from a common multi-screen application), namely first, second, and third windows 1.1, 1.1.1, and 1.1.1.1, previously positioned above the first desktop D1 on the primary screen 104, have now been moved to the

secondary screen 108 to uncover the first desktop D1 (Fig. 10C). Stated another way, the stack corresponding to the secondary screen 104 has from top-to-bottom the third window 1.1.1.1, the second window 1.1.1, the first window 1.1, and the second desktop D2. In the absence of the flick gesture, the drag gesture would have moved the third window 1.1.1.1 to the secondary screen 108 to reveal the second window 1.1.1 in the primary screen 104 (view not shown).

[0132] Although drag and flick gestures are depicted in Fig. 7-8, it is to be understood that different gestures may be used.

[0133] Desktop navigation is normally effected using a suitable gesture, such as a drag or flick gesture, received in the touch sensitive display. Unlike desktop navigation, gestures received by a gesture capture region 120 or 124 are used to move windows from one screen to the other (e.g., from one active or inactive position to another).

[0134] The exemplary systems and methods of this disclosure have been described in relation to communication devices. However, to avoid unnecessarily obscuring the present disclosure, the preceding description omits a number of known structures and devices. This omission is not to be construed as a limitation of the scopes of the claims. Specific details are set forth to provide an understanding of the present disclosure. It should however be appreciated that the present disclosure may be practiced in a variety of ways beyond the specific detail set forth herein.

[0135] Furthermore, while the exemplary aspects, embodiments, and/or configurations illustrated herein show the various components of the system collocated, certain components of the system can be located remotely, at distant portions of a distributed network, such as a LAN and/or the Internet, or within a dedicated system. Thus, it should be appreciated, that the components of the system can be combined in to one or more devices, such as a communication device, or collocated on a particular node of a distributed network, such as an analog and/or digital telecommunications network, a packet-switch network, or a circuit-switched network. It will be appreciated from the preceding description, and for reasons of computational efficiency, that the components of the system can be arranged at any location within a distributed network of components without affecting the operation of the system. For example, the various components can be located in a switch such as a PBX and media server, gateway, in one or more communications devices, at one or more users' premises, or some combination thereof.

Similarly, one or more functional portions of the system could be distributed between a telecommunications device(s) and an associated computing device.

[0136] Furthermore, it should be appreciated that the various links connecting the elements can be wired or wireless links, or any combination thereof, or any other known or later developed element(s) that is capable of supplying and/or communicating data to and from the connected elements. These wired or wireless links can also be secure links and may be capable of communicating encrypted information. Transmission media used as links, for example, can be any suitable carrier for electrical signals, including coaxial cables, copper wire and fiber optics, and may take the form of acoustic or light waves, such as those generated during radio-wave and infra-red data communications.

[0137] Also, while the flowcharts have been discussed and illustrated in relation to a particular sequence of events, it should be appreciated that changes, additions, and omissions to this sequence can occur without materially affecting the operation of the disclosed embodiments, configuration, and aspects.

[0138] A number of variations and modifications of the disclosure can be used. It would be possible to provide for some features of the disclosure without providing others.

[0139] For example in one alternative embodiment, the desktops are moved by gestures received in a gesture capture region 120, 124, and windows are moved by gestures received in a touch sensitive display 110, 114.

[0140] In yet another embodiment, the systems and methods of this disclosure can be implemented in conjunction with a special purpose computer, a programmed microprocessor or microcontroller and peripheral integrated circuit element(s), an ASIC or other integrated circuit, a digital signal processor, a hard-wired electronic or logic circuit such as discrete element circuit, a programmable logic device or gate array such as PLD, PLA, FPGA, PAL, special purpose computer, any comparable means, or the like. In general, any device(s) or means capable of implementing the methodology illustrated herein can be used to implement the various aspects of this disclosure. Exemplary hardware that can be used for the disclosed embodiments, configurations and aspects includes computers, handheld devices, telephones (e.g., cellular, Internet enabled, digital, analog, hybrids, and others), and other hardware known in the art. Some of these devices include processors (e.g., a single or multiple microprocessors), memory, nonvolatile storage, input devices, and output devices. Furthermore, alternative software

implementations including, but not limited to, distributed processing or component/object distributed processing, parallel processing, or virtual machine processing can also be constructed to implement the methods described herein.

[0141] In yet another embodiment, the disclosed methods may be readily implemented in conjunction with software using object or object-oriented software development environments that provide portable source code that can be used on a variety of computer or workstation platforms. Alternatively, the disclosed system may be implemented partially or fully in hardware using standard logic circuits or VLSI design. Whether software or hardware is used to implement the systems in accordance with this disclosure is dependent on the speed and/or efficiency requirements of the system, the particular function, and the particular software or hardware systems or microprocessor or microcomputer systems being utilized.

[0142] In yet another embodiment, the disclosed methods may be partially implemented in software that can be stored on a storage medium, executed on programmed general-purpose computer with the cooperation of a controller and memory, a special purpose computer, a microprocessor, or the like. In these instances, the systems and methods of this disclosure can be implemented as program embedded on personal computer such as an applet, JAVA® or CGI script, as a resource residing on a server or computer workstation, as a routine embedded in a dedicated measurement system, system component, or the like. The system can also be implemented by physically incorporating the system and/or method into a software and/or hardware system.

[0143] Although the present disclosure describes components and functions implemented in the aspects, embodiments, and/or configurations with reference to particular standards and protocols, the aspects, embodiments, and/or configurations are not limited to such standards and protocols. Other similar standards and protocols not mentioned herein are in existence and are considered to be included in the present disclosure. Moreover, the standards and protocols mentioned herein and other similar standards and protocols not mentioned herein are periodically superseded by faster or more effective equivalents having essentially the same functions. Such replacement standards and protocols having the same functions are considered equivalents included in the present disclosure.

[0144] The present disclosure, in various aspects, embodiments, and/or configurations, includes components, methods, processes, systems and/or apparatus substantially as depicted and

described herein, including various aspects, embodiments, configurations embodiments, subcombinations, and/or subsets thereof. Those of skill in the art will understand how to make and use the disclosed aspects, embodiments, and/or configurations afrer understanding the present disclosure. The present disclosure, in various aspects, embodiments, and/or configurations, includes providing devices and processes in the absence of items not depicted and/or described herein or in various aspects, embodiments, and/or configurations hereof, including in the absence of such items as may have been used in previous devices or processes, e.g., for improving performance, achieving ease and\or reducing cost of implementation.

[0145] The foregoing discussion has been presented for purposes of illustration and description. The foregoing is not intended to limit the disclosure to the form or forms disclosed herein. In the foregoing Detailed Description for example, various features of the disclosure are grouped together in one or more aspects, embodiments, and/or configurations for the purpose of streamlining the disclosure. The features of the aspects, embodiments, and/or configurations of the disclosure may be combined in alternate aspects, embodiments, and/or configurations other than those discussed above. This method of disclosure is not to be interpreted as reflecting an intention that the claims require more features than are expressly recited in each claim. Rather, as the following claims reflect, inventive aspects lie in less than all features of a single foregoing disclosed aspect, embodiment, and/or configuration. Thus, the following claims are hereby incorporated into this Detailed Description, with each claim standing on its own as a separate preferred embodiment of the disclosure.

[0146] Moreover, though the description has included description of one or more aspects, embodiments, and/or configurations and certain variations and modifications, other variations, combinations, and modifications are within the scope of the disclosure, e.g., as may be within the skill and knowledge of those in the art, afrer understanding the present disclosure. It is intended to obtain rights which include alternative aspects, embodiments, and/or configurations to the extent permitted, including alternate, interchangeable and/or equivalent structures, functions, ranges or steps to those claimed, whether or not such alternate, interchangeable and/or equivalent structures, functions, ranges or steps are disclosed herein, and without intending to publicly dedicate any patentable subject matter.

What is claimed is:

1. 1. A method, comprising:

receiving, by at least one of a gesture capture region and a touch sensitive display, a first gesture; and

in response, moving, by processor-executable middleware, a first displayed image on a first touch sensitive display of a communication device to a second touch sensitive display of the communication device;

while the first displayed image is being moved to the second touch sensitive display, receiving, by the at least one of a gesture capture region and a touch sensitive display, a second gesture;

in response to the second gesture, retracting, by the processor-executable middleware, movement of the first displayed image and moving, by the processor-executable middleware, at least a second displayed image to the second touch sensitive display, wherein the first and second displayed images are at least one of a desktop and window of an application.

2. The method of claim 1, wherein the first and second displayed images are windows, wherein the first gesture is a drag, and wherein the second gesture is a flick.

3. The method of claim 1, wherein the first and second displayed images are controlled by a common multi-screen application and wherein, in response to the second gesture, both the first and second displayed images are moved from the first touch sensitive display to the second touch sensitive display.

4. The method of claim 1, wherein the first gesture is operable to move only the first and not the second displayed image from the first touch sensitive display to the second touch sensitive display.

5. The method of claim 1, wherein the first and second gestures are received by a common gesture capture region and wherein the gesture capture region is unable to display any displayed image.

6. The method of claim 1, wherein the first and second displayed images are windows controlled by a common multi-screen application.

7. A non-transient computer readable medium comprising processor executable instructions operable to perform the steps of claim 1.

2. 8. A method, comprising:

receiving, by a gesture capture region, a drag gesture;

in response, moving, by processor-executable middleware, a first window from a first touch sensitive display to a second touch sensitive display;

while the first window is moving from the first to the second touch sensitive display, receiving, by the gesture capture region, a flick gesture; and

in response, retracting, by the processor-executable middleware, movement of the first window and moving, by the processor-executable middleware, a second window from the first to the second touch sensitive display.

9. The method of claim 8, wherein the first and second windows are controlled by a common multi-screen application and wherein, in response to the flick gesture, both the first and second windows are moved from the first touch sensitive display to the second touch sensitive display.

10. The method of claim 8, wherein the drag gesture is operable to move only the first and not the second window from the first touch sensitive display to the second touch sensitive display.

11. The method of claim 8, wherein the drag and flick gestures are received by a common gesture capture region and wherein the gesture capture region is unable to display any displayed image and wherein the first and second displayed images are windows controlled by a common multi-screen application.

12. A non-transient computer readable medium comprising processor executable instructions operable to perform the steps of claim 8.

13. A dual display communication device, comprising:

a gesture capture region to receive a flick gesture; and

a first touch sensitive display to receive a flick gesture and display displayed images, wherein the displayed images are at least one of a desktop and window of an application;

a second touch sensitive display to receive a flick gesture and display displayed images;

middleware operable to perform at least one of the following operations:

in response to a first gesture, move a first displayed image on the first touch sensitive display to the second touch sensitive display;

while the first displayed image is being moved to the second touch sensitive display, receive notification of a second gesture;

in response to receipt of notification of the second gesture, retract movement of the first displayed image and move at least a second displayed image to the second touch sensitive display, wherein the first and second displayed images are at least one of a desktop and window of an application.

14. The device of claim 13, wherein the first and second displayed images are windows, wherein the first gesture is a drag, and wherein the second gesture is a flick.

15. The device of claim 13, wherein the first and second displayed images are controlled by a common multi-screen application and wherein, in response to the second gesture, both the first and second displayed images are moved from the first touch sensitive display to the second touch sensitive display.

16. The device of claim 13, wherein the first gesture is operable to move only the first and not the second displayed image from the first touch sensitive display to the second touch sensitive display.

17. The device of claim 13, wherein the first and second gestures are received by a common gesture capture region and wherein the gesture capture region is unable to display any displayed image.

[0001] 18. The device of claim 13, wherein the first and second displayed images are windows controlled by a common multi-screen application.

ABSTRACT

[0147] The disclosed method and device are directed to navigation, by a dual display communication device, through display objects.

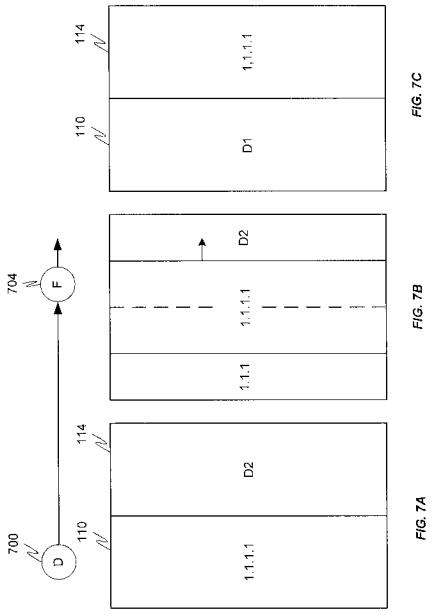


FIG. 7C

FIG. 7B

FIG. 7A

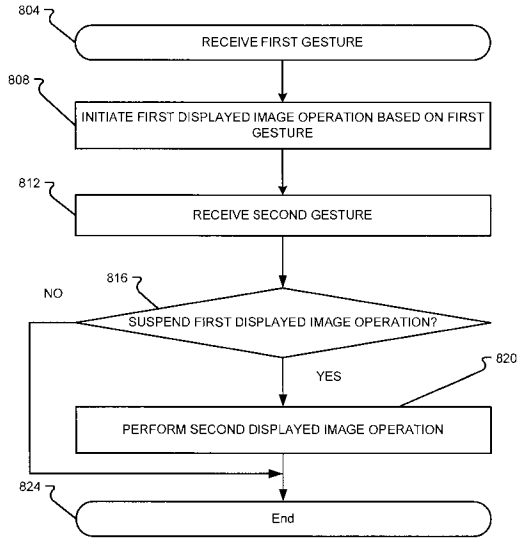


FIG. 8

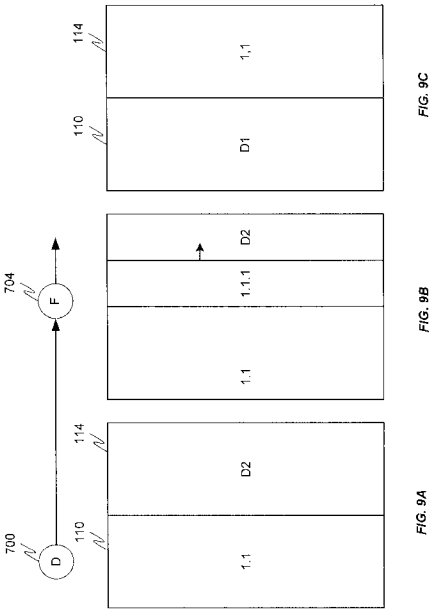


FIG. 9C

FIG. 9B

FIG. 9A

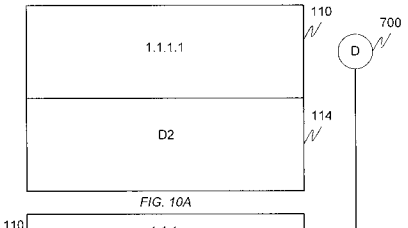


FIG. 10A

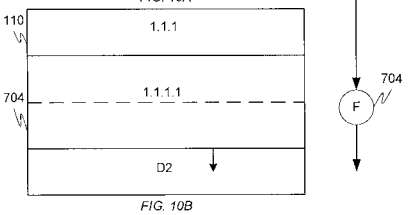


FIG. 10B

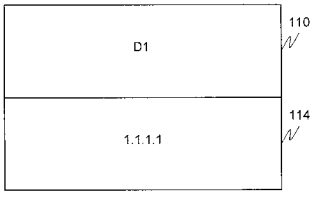


FIG. 10C

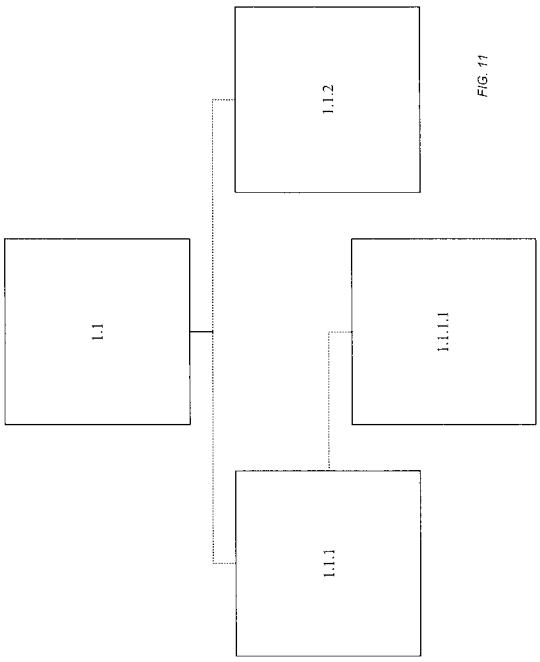


FIG. 11

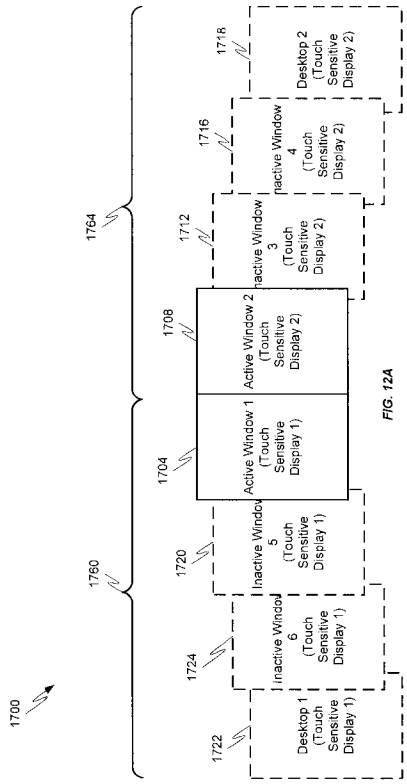


FIG. 12A

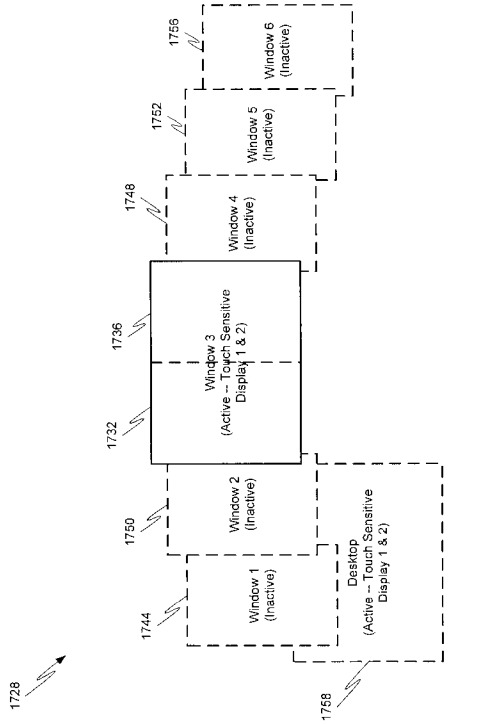


FIG. 12B

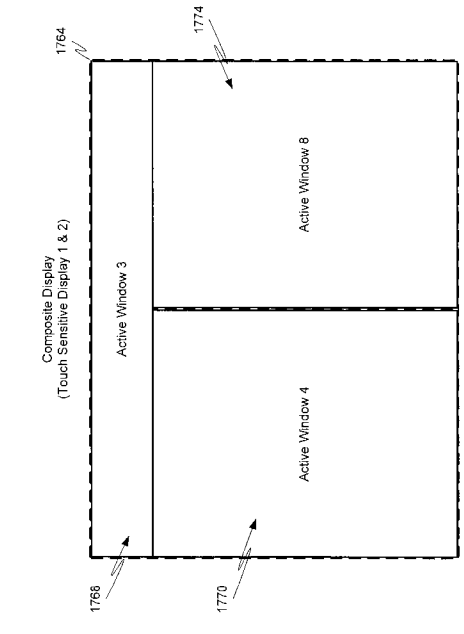


FIG. 12C

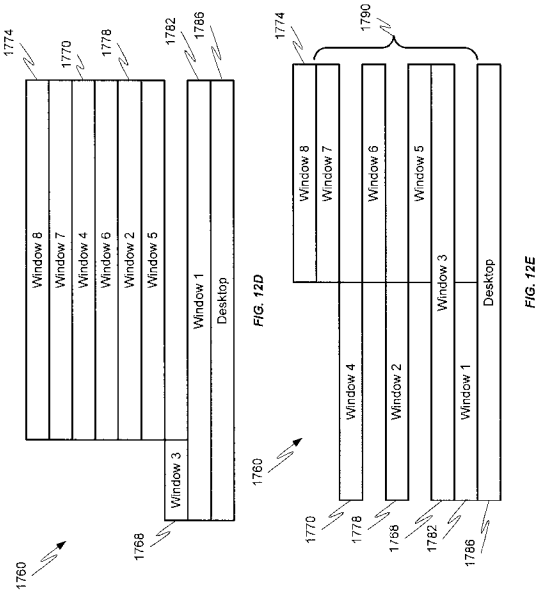


FIG. 12D

FIG. 12E

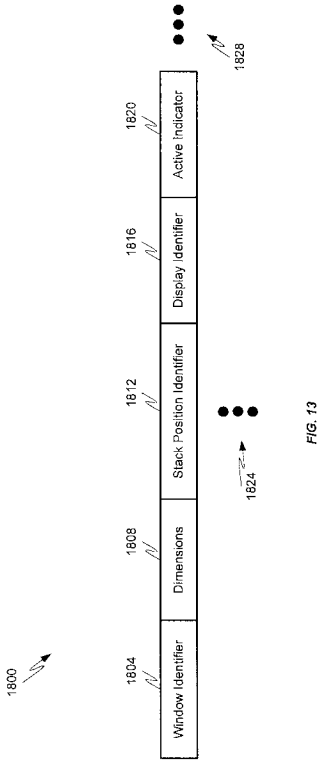


FIG. 13

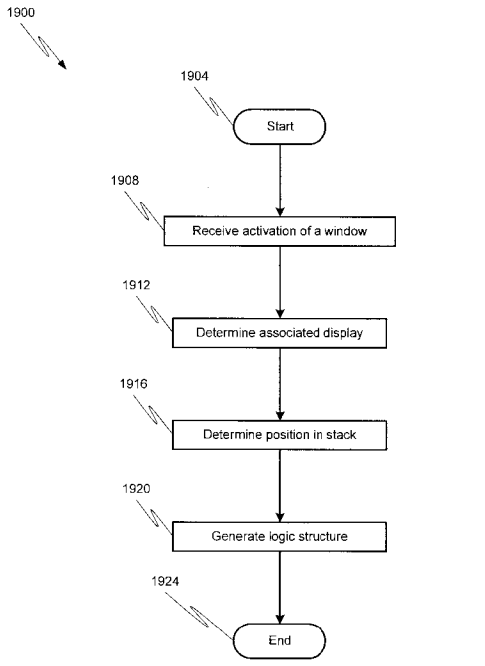


FIG. 14

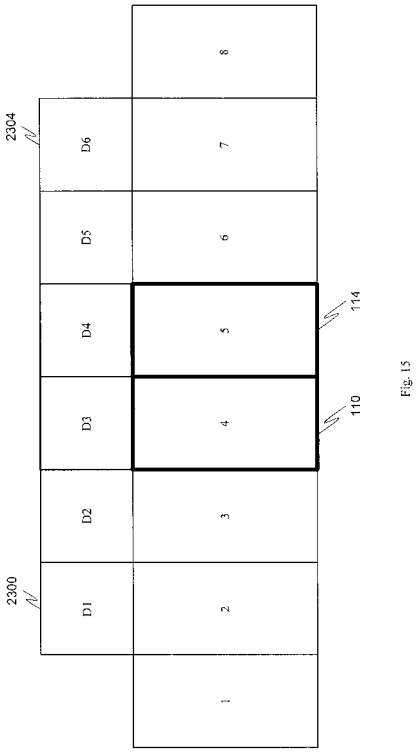


Fig. 15

OFF-SCREEN GESTURE DISMISSABLE KEYBOARD

[0100] Fig. 7A depicts a device 100 with first 104 and second 108 screens. In this example, the screens 104, 108 are each in a portrait mode. However, embodiments of the invention disclosed herein are equally applicable to use in connection with a display 104, 108 that is in a landscape orientation.

[0101] In the state depicted in Fig. 7A, the first screen 104 displays a first set of information, for example a window comprising a first application 708. The second screen 108 displays a second set of information, for example a window comprising a second application 712. In addition, a virtual keyboard 704 is presented by the screens 104, 108. In this example, where the screens 104, 108 are in a portrait mode, the virtual keyboard 704 spans portions of both screens 104, 108. As can be appreciated by one of skill in the art after consideration of the present disclosure, where the screens 104, 108 are in a landscape orientation, a virtual keyboard 704 can be displayed by one screen 104, 108 only. In general, the virtual keyboard 704 contains a plurality of keys or buttons to enable input by a user. As an example, and without limitation, the virtual keyboard 704 can comprise a conventional alphanumeric keyboard. Moreover, the virtual keyboard 704 is presented by portions of the screens 104, 108 comprising touch sensitive display 110, 114 areas or regions of the screens 104, 108.

[0102] The gesture 714 depicted in Fig. 7A may comprise a gesture or other touch input by a user that is entered in an area or region of the screens 104, 108 that is outside of the area or region in which the virtual keyboard 704 is displayed. In this example, the gesture 714 is received in the gesture capture region 124 of the second screen 108. Moreover, the gesture 714 can comprise a window management gesture. For example, in accordance with at least some embodiments of the present invention, the gesture 714, can be entered in a gesture capture region 120 and/or 124 of a touch sensitive screen 104 and/or 108. Although the specific gesture 714 illustrated in the figure comprises a drag gesture 400, other gestures or inputs entered outside of the portion of the screen or screens 104 and/or 108 displaying the virtual keyboard 704 can be used.

[0103] Fig. 7B depicts the output of the screens 104, 108 after the receipt of the gesture 400, 714 from the user. In particular, the screens 104, 108 no longer display the keyboard 704. That is, the keyboard 704 has been dismissed in response to the receipt of the gesture 400, 714. In addition, the command entered as part of the gesture 400, 714 is executed. Thus, in this example, where the gesture 714 input is a window management operation moving a window including the second application 712, after receipt of the gesture 714 the window comprising the second application 712 is now displayed by the first screen 104, while a desktop 716 is revealed on the second screen 108.

[0104] Fig. 8A depicts an output configuration similar to that shown in Fig. 7A. However, in this example, the gesture 400, 714 comprises a different command. In this example, the gesture 714 is received in the gesture capture region 120 of the first screen. The result of the receipt of this gesture 714 is shown in Fig. 8B. In particular, here the gesture is a window management gesture or input that moves the window that had been displayed by the first screen 104 to the second screen 108, revealing a desktop 720 on the first screen 104 (*see* Fig. 8B). Moreover, as a result of the receipt of the gesture input 400, 714 acting on or with respect to the first window 708, the virtual keyboard 704 is dismissed.

[0105] Fig. 9A depicts another exemplary output of the screens 104, 108 of the device 100 in accordance with embodiments of the present invention. In this example, a root or parent window application 724 is presented by the first screen 104, while the second desktop 716 is displayed on the second screen 108. In addition, a virtual keyboard 704 spanning both screens 104, 108 is provided. In this example, a gesture 400, 714 comprising a window management input is entered with respect to the root window of the first application 724 through a gesture entered in the gesture capture region 120 of the first screen 104. The result of this input is depicted in Fig. 9B. In particular, as shown in Fig. 9B, the receipt of the input causes a child application window 728 to be presented on the second screen 108. In addition, the entry of the input causes the virtual keyboard 704 to be dismissed. Accordingly, a larger area of the screens 104, 108 is available to present the application windows 724, 728.

[0106] With reference now to Fig. 10, aspects of a method for dismissing a keyboard or discontinuing the presentation of a virtual keyboard 704 in accordance with embodiments of the present invention are depicted. Initially, at step 1004, a determination is made as to whether a virtual keyboard 704 is to be displayed. Until such a determination is made, the process may idle at step 1004. After a determination to display a virtual keyboard 704 has been made, output and a virtual keyboard 704 are provided using the first 104 and second 108 screens of the device 100 (step 1008). As can be appreciated by one of skill in the art after consideration of the present disclosure, the use of first and second screens 104, 108 does not require that the virtual keyboard

704 have portions presented by both the first 104 and second 108 screens. Instead, as an alternative to being presented across portions of first 104 and second 108 screens, a virtual keyboard 704 can be displayed by one of the first 104 and second 108 screens only.

[0107] At step 1012, a determination is made as to whether input has been received from a user in an area of the virtual keyboard 704. As an example, input received can include a press of a virtual key included within the virtual keyboard 704. If input from the user in the area corresponding to the displayed virtual keyboard 704 is received, that user input is processed normally (step 1016). In particular, the action indicated by the key press is performed. As examples, and without limitation, such actions can include the entry of text, numbers or commands.

[0108] After determining that user input has not been received within the area of the virtual keyboard 704, or after processing that input, a determination can be made as to whether input from the user has been received in an area of the screens 104, 108 outside of the area occupied by the virtual keyboard 704. Such input can be input received within any touch sensitive region of the screens 104, 108. For example, the input can be received within a touch sensitive display region 110, 114, a configurable area 112, 116, and/or a gesture capture region 120, 124 of a screen 104, 108. In accordance with other embodiments, the input is required to be a window management input to control a display on a screen 104, 108. In accordance with further embodiments, the input can be required to be received within a gesture capture region 120, 124. If input is received from a user in an area of a screen 104 and/or 108 outside of the area presenting the virtual keyboard 704, the action indicated by the input can be performed, and presentation of the virtual keyboard 704 is discontinued (step 1024). As an example, and without limitation, input received in an area of a screen 104, 108 outside of an area presenting a virtual keyboard 704 can comprise a command to change the item within a stack of items displayed by a screen 104, 108, or to otherwise manage windows presented by the screens 104, 108. Moreover, although such action is directed to the management of windows or elements presented by a screen 104, 108, embodiments of the disclosed invention, in response to the receipt of such input, discontinues the presentation of the virtual keyboard 704.

[0109] After a determination that input from a user in an area outside of the virtual keyboard 704 has not been received, or after receiving such input, performing an indicated action and discontinuing the presentation of the virtual keyboard 704, a determination can be made as to

whether the device 100 has been powered off (step 1028). If the device has not been powered off, the process can return to step 1004, to determine whether a virtual keyboard 704 should be displayed. If the device 100 has been powered off, the process ends.

[0110] In accordance with embodiments of the present invention, the method disclosed herein can be performed by the execution of application programming stored in memory 208, 508 and by a processor 204, 504. For instance, a windows management module or class 532 can include functionality to receive and act on input received from a user. Moreover, such input can include window management input, and can include the execution of steps to discontinue the presentation of a virtual keyboard 704 in response to such input. A windows management module or class 532 or other middleware 520 can also provide the virtual keyboard 704 and receive input provided by a user in connection with the virtual keyboard 704. In addition, although certain embodiments have been described in connection with operation on a device having first 104 and second 108 screens, the invention is not limited to operation on such a device. For example, a device 100 can have more than two screens, and/or the screen can comprise virtual screens or windows.

[0111] The exemplary systems and methods of this disclosure have been described in relation to dismissing a virtual keyboard 704 or discontinuing the presentation of a virtual keyboard 704 in response to receiving input from a user outside of an area of the touch screen display. However, to avoid unnecessarily obscuring the present disclosure, the preceding description omits a number of known structures and devices. This omission is not to be construed as a limitation of the scopes of the claims. Specific details are set forth to provide an understanding of the present disclosure. It should however be appreciated that the present disclosure may be practiced in a variety of ways beyond the specific detail set forth herein.

[0112] Furthermore, while the exemplary aspects, embodiments, and/or configurations illustrated herein show the various components of the system collocated, certain components of the system can be located remotely, at distant portions of a distributed network, such as a LAN and/or the Internet, or within a dedicated system. Thus, it should be appreciated, that the components of the system can be combined in to one or more devices, such as a computer, laptop, netbook, tablet computer, smart phone, mobile device, etc., or collocated on a particular node of a distributed network, such as an analog and/or digital telecommunications network, a packet-switch network, or a circuit-switched network. It will be appreciated from the preceding

description, and for reasons of computational efficiency, that the components of the system can be arranged at any location within a distributed network of components without affecting the operation of the system. For example, the various components can be located in a switch such as a PBX and media server, gateway, in one or more communications devices, at one or more users' premises, or some combination thereof. Similarly, one or more functional portions of the system could be distributed between a telecommunications device(s) and an associated computing device.

[0113] Furthermore, it should be appreciated that the various links connecting the elements can be wired or wireless links, or any combination thereof, or any other known or later developed element(s) that is capable of supplying and/or communicating data to and from the connected elements. These wired or wireless links can also be secure links and may be capable of communicating encrypted information. Transmission media used as links, for example, can be any suitable carrier for electrical signals, including coaxial cables, copper wire and fiber optics, and may take the form of acoustic or light waves, such as those generated during radio-wave and infra-red data communications.

[0114] Also, while the flowcharts have been discussed and illustrated in relation to a particular sequence of events, it should be appreciated that changes, additions, and omissions to this sequence can occur without materially affecting the operation of the disclosed embodiments, configuration, and aspects.

[0115] A number of variations and modifications of the disclosure can be used. It would be possible to provide for some features of the disclosure without providing others.

[0116] For example in one alternative embodiment, the virtual keyboard 704 can be displayed within only one screen 104, 108.

[0117] In another alternative embodiment, a virtual keyboard 704 can be displayed as part of one or more touch sensitive displays 110, 114, and a gesture 714 can be received in a gesture capture region 120, 124.

[0118] In yet another embodiment, the systems and methods of this disclosure can be implemented in conjunction with a special purpose computer, a programmed microprocessor or microcontroller and peripheral integrated circuit element(s), an ASIC or other integrated circuit, a digital signal processor, a hard-wired electronic or logic circuit such as discrete element circuit, a programmable logic device or gate array such as PLD, PLA, FPGA, PAL, special purpose

computer, any comparable means, or the like. In general, any device(s) or means capable of implementing the methodology illustrated herein can be used to implement the various aspects of this disclosure. Exemplary hardware that can be used for the disclosed embodiments, configurations and aspects includes computers, handheld devices, telephones (e.g., cellular, Internet enabled, digital, analog, hybrids, and others), and other hardware known in the art. Some of these devices include processors (e.g., a single or multiple microprocessors), memory, nonvolatile storage, input devices, and output devices. Furthermore, alternative software implementations including, but not limited to, distributed processing or component/object distributed processing, parallel processing, or virtual machine processing can also be constructed to implement the methods described herein.

[0119] In yet another embodiment, the disclosed methods may be readily implemented in conjunction with software using object or object-oriented software development environments that provide portable source code that can be used on a variety of computer or workstation platforms. Alternatively, the disclosed system may be implemented partially or fully in hardware using standard logic circuits or VLSI design. Whether software or hardware is used to implement the systems in accordance with this disclosure is dependent on the speed and/or efficiency requirements of the system, the particular function, and the particular software or hardware systems or microprocessor or microcomputer systems being utilized.

[0120] In yet another embodiment, the disclosed methods may be partially implemented in software that can be stored on a storage medium, executed on programmed general-purpose computer with the cooperation of a controller and memory, a special purpose computer, a microprocessor, or the like. In these instances, the systems and methods of this disclosure can be implemented as program embedded on personal computer such as an applet, JAVA® or CGI script, as a resource residing on a server or computer workstation, as a routine embedded in a dedicated measurement system, system component, or the like. The system can also be implemented by physically incorporating the system and/or method into a software and/or hardware system.

[0121] Although the present disclosure describes components and functions implemented in the aspects, embodiments, and/or configurations with reference to particular standards and protocols, the aspects, embodiments, and/or configurations are not limited to such standards and protocols. Other similar standards and protocols not mentioned herein are in existence and are

considered to be included in the present disclosure. Moreover, the standards and protocols mentioned herein and other similar standards and protocols not mentioned herein are periodically superseded by faster or more effective equivalents having essentially the same functions. Such replacement standards and protocols having the same functions are considered equivalents included in the present disclosure.

[0122] The present disclosure, in various aspects, embodiments, and/or configurations, includes components, methods, processes, systems and/or apparatus substantially as depicted and described herein, including various aspects, embodiments, configurations embodiments, subcombinations, and/or subsets thereof. Those of skill in the art will understand how to make and use the disclosed aspects, embodiments, and/or configurations afrer understanding the present disclosure. The present disclosure, in various aspects, embodiments, and/or configurations, includes providing devices and processes in the absence of items not depicted and/or described herein or in various aspects, embodiments, and/or configurations hereof, including in the absence of such items as may have been used in previous devices or processes, e.g., for improving performance, achieving ease and\or reducing cost of implementation.

[0123] The foregoing discussion has been presented for purposes of illustration and description. The foregoing is not intended to limit the disclosure to the form or forms disclosed herein. In the foregoing Detailed Description for example, various features of the disclosure are grouped together in one or more aspects, embodiments, and/or configurations for the purpose of streamlining the disclosure. The features of the aspects, embodiments, and/or configurations of the disclosure may be combined in alternate aspects, embodiments, and/or configurations other than those discussed above. This method of disclosure is not to be interpreted as reflecting an intention that the claims require more features than are expressly recited in each claim. Rather, as the following claims reflect, inventive aspects lie in less than all features of a single foregoing disclosed aspect, embodiment, and/or configuration. Thus, the following claims are hereby incorporated into this Detailed Description, with each claim standing on its own as a separate preferred embodiment of the disclosure.

[0124] Moreover, though the description has included description of one or more aspects, embodiments, and/or configurations and certain variations and modifications, other variations, combinations, and modifications are within the scope of the disclosure, e.g., as may be within the skill and knowledge of those in the art, after understanding the present disclosure. It is intended

to obtain rights which include alternative aspects, embodiments, and/or configurations to the extent permitted, including alternate, interchangeable and/or equivalent structures, functions, ranges or steps to those claimed, whether or not such alternate, interchangeable and/or equivalent structures, functions, ranges or steps are disclosed herein, and without intending to publicly dedicate any patentable subject matter.

What is claimed is:

1. A method controlling an input, comprising:

presenting by a touch screen display of a device a set of virtual keys;

receiving a first input from a user, wherein the first input is received outside of an area of the touch screen display;

in response to the received first input, discontinuing the presentation of the set of virtual keys by the touch screen display of the device.

2. The method of claim 1, wherein the first input is received in a gesture capture region, and wherein the gesture capture region is included as part of a screen of the device.

3. The method of claim 2, wherein the touch screen display is included as part of a screen of the device, and wherein the gesture capture region of the screen is outside of an area of the screen in which the touch screen display is presented.

4. The method of claim 3, wherein a first portion of the set of virtual keys is presented by a first touch screen included as part of a first screen of the device, wherein a second portion of the set of virtual keys is presented by a second touch screen included as part of a second screen of the device, and wherein the gesture capture region is included as part of one of the first screen of the device and the second screen of the device.

5. The method of claim 3, wherein a first portion of the set of virtual keys is presented by a first touch screen included as part of a first screen of the device, wherein a second portion of the set of virtual keys is presented by a second touch screen included as part of a second screen of the device, wherein the first input is partially received within a first gesture capture region included as part of the first screen of the device, and wherein the first input is partially received within a second gesture capture region included as part of the second screen of the device.

6. The method of claim 1, further comprising:

presenting at least a first portion of first information, wherein at least a first portion of the first information is presented simultaneously with the presentation of the set of virtual keys.

7. The method of claim 6, further comprising:

after discontinuing the presentation of the set of virtual keys by a touch screen display of the device, presenting at least a second portion of the first information in addition to the first portion of the first information.

8. The method of claim 6, wherein at least a part of the first portion of first information is presented by a first touch screen included as part of a first screen of the device, wherein at least another part of the first portion of first information is presented by a second touch screen included as part of a second screen of the device, wherein a first portion of the set of virtual keys is presented by the first touch screen, wherein a second portion of the set of virtual keys is presented by the second touch screen, wherein the first input is received within at least one of a first gesture capture region included as part of the first screen of the device but outside of an area of the first touch screen and a second gesture capture region included as part of the second screen of the device but outside of an area of the second touch screen.

9. The method of claim 1, wherein the received first input is a window management input.

10. The method of claim 9, wherein the received first input includes one of a drag gesture, a flick gesture, a tap, a long press, a pin and drag, and a pinch gesture.

11. A device, comprising:

a screen, including:

a touch screen display, wherein the touch screen display is located within a first area of the screen;

a gesture capture region, wherein the gesture capture region is located within a second area of the screen;

memory;

a processor;

application programming stored in the memory and executed by the processor, wherein the application programming is operable to:

display a set of virtual keys using the touch screen display;

receive an input from a user entered in the gesture capture region;

in response to the input received from the user entered in the gesture capture region, discontinue displaying a set of virtual keys.

12. The device of claim 11, wherein the gesture capture region is not operative to display output to the user.

13. The device of claim 11, wherein the first and second areas of the screen do not overlap.

14. The device of claim 11, wherein the screen includes first and second screens, wherein the first screen includes:

a first touch screen display, wherein the first touch screen display is located within a first area of the first screen;

a first gesture capture region, wherein the first gesture capture region is located within a second area of the first screen,

wherein the second screen includes:

a second touch screen display, wherein the second touch screen display is located within a second area of the first screen;

a second gesture capture region, wherein the second gesture capture region is located within a second area of the second screen,

15. The device of claim 14, wherein the set of virtual keys is displayed using a first portion of the first touch screen display and a first portion of the second touch screen display.

16. The device of claim 15, wherein first information is displayed using a second portion of the first touch screen display and a second portion of the second touch screen display.

17. A computer readable medium having stored thereon computer-executable instructions, the computer executable instructions causing a processor to execute a method for dismissing virtual key sets, the computer-executable instructions comprising:

instructions to display a first set of virtual keys on a touch screen;

instructions to determine whether a user input has been received within a gesture capture region;

instructions to discontinue the display of the first set of virtual keys on the touch screen in response to a determination that user input has been received within the gesture capture region.

18. The computer readable medium of claim 17, wherein the touch screen is part of at least a first screen, and wherein the gesture capture region is part of at least the first screen.

19. The computer readable medium of claim 17, wherein the touch screen includes a first touch screen included as part of a first screen and a second touch screen included as part of a second screen, and wherein the gesture capture region includes a first gesture capture region included as part of the first screen and a second gesture capture region included as part of the second screen.

20. The computer readable medium of claim 19, wherein the display of the first set of virtual keys is discontinued in response to user input is received within one of the first gesture capture region and the second gesture capture region.

ABSTRACT

Methods and devices for discontinuing the presentation of a virtual keyboard or dismissing a virtual keyboard in response to the receipt of input are provided. More particularly, input entered outside of an area of a screen comprising a touch screen display can cause the presentation of a virtual keyboard to be discontinued. The discontinuance of the virtual keyboard display can be performed simultaneously with an operation indicated by the received input.

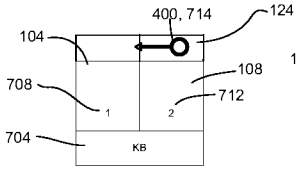


Fig. 7A

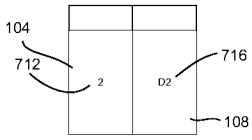


Fig. 7B

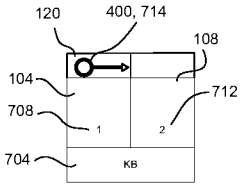


Fig. 8A

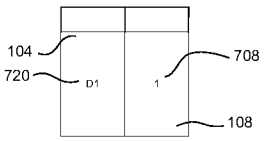


Fig. 8B

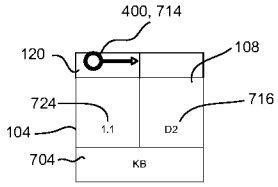


Fig. 9A

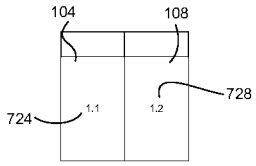


Fig. 9B

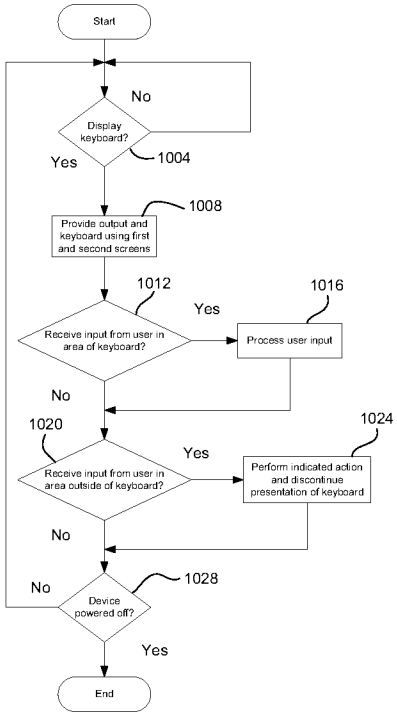


Fig. 10

KEYBOARD SPANNING MULTIPLE SCREENS

[0100] Fig. 7A depicts a device 100 with first 104 and second 108 screens. In this example, the screens 104, 108 are in portrait mode. A virtual keyboard 704 is presented by the screens 104, 108. In particular, a first portion 704a of the virtual keyboard 704 is presented by a first portion of the first screen 104, while a second portion 704b of the virtual keyboard 704 is presented by a first portion of the second screen 108. More particularly, the virtual keyboard 704 spans a line of separation 708 between the first and second screens. Moreover, the virtual keyboard 704 does not occupy all of the area of either the first screen 104 or the second screen 108. Accordingly, a second portion of the first screen 104 and a second portion of the second screen 108 remain available for the presentation of information other than the virtual keyboard 704 to a user.

[0101] Fig. 7B depicts a device 100 with first 104 and second 108 screens in a landscape mode. In this example, the first screen 104 is above and adjacent to the second screen 108. In addition, a virtual keyboard 704 is presented by the second screen 108. As shown, the virtual keyboard 704 can occupy less than all of the area of the second screen 108. In accordance with other embodiments, the virtual keyboard 704 can occupy the entire area of the second screen 108. In addition, the area occupied by the virtual keyboard 704 can be the same or about the same (*e.g.*, within ±10%) while the screens 104, 108 are in either portrait or landscape modes. In accordance with still other embodiments, the virtual keyboard 704 can be presented by the first screen 104 while in a landscape mode. For example, where the device 100 is held or positioned such that the second screen 108 is above the first screen 104, the virtual keyboard 704 can be presented by the first screen 104. In accordance with still further embodiments of the present invention, a virtual keyboard 704 can be presented using portions of the first 104 and second 108 screens, even when those screens are in a landscape mode.

[0102] Fig. 8A presents details of an exemplary virtual keyboard 704, with a first portion 704a presented using a first portion of a first screen 104 and with a second portion 704b presented using a first portion of a second screen 108 of a device 100 in accordance with embodiments of the present invention. As shown in this example, and without limitation, the virtual keyboard 704 can comprise a QWERTY keyboard. Because the virtual keyboard 704 is presented using two adjacent screens 104, 108, the line of separation 708 passes between a first portion 704a of the virtual keyboard 704 presented using the first screen 104, and a second portion 704b of the virtual keyboard 704 presented by the second screen 108. Moreover, because of the line of separation 708, the spacing of the keys 816 of the virtual keyboard 704 can be arranged such that the line of separation 708 falls between individual character keys 816. A space bar 820 can span across both the first 104 and second 108 screens.

[0103] Fig. 8B depicts a virtual keyboard 704 displayed by one of the screens 104 or 108 when the screens 104, 108 are in a landscape mode. In particular, in this example, the virtual keyboard 704 is displayed by the second screen 108, while the area of the first screen 104 remains available to present information other than the virtual keyboard 704 to a user.

[0104] Comparing the virtual keyboard 704 presented with portions of the screens 104, 108 in a portrait mode, such that the virtual keyboard 704 spans the two screens 104, 108, to the mode of operation in which the virtual keyboard 704 is presented by a single screen 104 or 108 in a landscape mode highlights changes that can be made to accommodate the line of separation 708. In particular, in the embodiment illustrated in Fig. 8A, the spacing of the keys 816 has been altered so that the smaller keys (*e.g.*, the character keys) lie on one side or the other of the line of separation 708. In contrast, in the embodiment illustrated in Fig. 8B, the keys 816 of the virtual keyboard 704 have a more conventional, interleaved type spacing.

[0105] Fig. 9 depicts aspects of a method for presenting a virtual keyboard 704 in accordance with embodiments of the present invention. Initially, at step 904, a determination is made as to whether a virtual keyboard is to be displayed. Until such a determination is made, the process may idle at step 904. After a determination to display a virtual keyboard 704 has been made, a determination can next be made as to the orientation of the device 100 (step 908). If the device 100 is in a portrait mode, the virtual keyboard 704 is presented using portions of both the first 104 and second 108 screens of the multiple screen device 100 (step 912). Accordingly, with the screens 104, 108 in the portrait mode, the virtual keyboard 704 spans portions of both screens 104, 108. If the device 100 screens 104, 108 are determined to be in a landscape orientation, the virtual keyboard 704 can be presented using one of the screens 104 or 108 (step 916). For example, the virtual keyboard 704 can be presented by the screen 104 or 108 that is below the other screen 104 or 108. After presenting the virtual keyboard 704 in the mode determined to be correct, a determination can be made as to whether the orientation of the device has changed (step 920). If the orientation has changed, the process can return to step 908. If the orientation has not changed, a determination can next be made as to whether the content of the displayed

output has changed (step 924). A change to the output can include the display of characters entered by a user through the virtual keyboard 704. In accordance with other embodiments, a change to the output can include a change to the information displayed through operation of application programming and/or the receipt of information at the device 100. A change in output can also include an acknowledgement of commands received from a user or from a connected device, or that are generated through operation of application programming, and the effect or processing of the command. After determining that the output provided by the displays 104, 108 has changed, the process can return to step 904, to determine whether, in view of the change or other information, the virtual keyboard 704 should continue to be displayed.

[0106] At step 928, a determination can be made as to whether operation of the device should continue, or whether the device has powered down. If operation is to continue, the process can return to step 904. If the device has been powered down, the process can end. Although steps in the process have been shown and described in a particular order, it should be appreciated that other orderings are possible and incorporated herein. For example, the process can be discontinued at any point in the process if the device 100 is powered off.

In accordance with embodiments of the present invention, the method disclosed herein can be performed by the execution of application programming or other software components stored in memory 208, 508 by a processor 204, 504. For instance, the operating system 516, a component or components of the framework 520, and/or a component or components of the multi-display management module 524 can include or provide functionality to receive and act on input received from a user in connection with a virtual keyboard 704. Moreover, such input can include input comprising user interface interactions by a user that are managed by the input management module 536, or other components. In addition, although certain embodiments have been described in connection with operation on a device having first 104 and second 108 screens, the invention is not limited to operation on such a device. For example, a device 100 can have more than two screens, and/or the screen can comprise virtual screens or windows.

[0107] The exemplary systems and methods of this disclosure have been described in relation to the presentation of a virtual keyboard 704 that spans multiple screens 104, 108. However, to avoid unnecessarily obscuring the present disclosure, the preceding description omits a number of known structures and devices. This omission is not to be construed as a limitation of the scopes of the claims. Specific details are set forth to provide an understanding of the present

disclosure. It should however be appreciated that the present disclosure may be practiced in a variety of ways beyond the specific detail set forth herein.

[0108] Furthermore, while the exemplary aspects, embodiments, and/or configurations illustrated herein show the various components of the system collocated, certain components of the system can be located remotely, at distant portions of a distributed network, such as a LAN and/or the Internet, or within a dedicated system. Thus, it should be appreciated, that the components of the system can be combined in to one or more devices, such as a computer, laptop, netbook, tablet computer, smart phone, mobile device, etc., or collocated on a particular node of a distributed network, such as an analog and/or digital telecommunications network, a packet-switch network, or a circuit-switched network. It will be appreciated from the preceding description, and for reasons of computational efficiency, that the components of the system can be arranged at any location within a distributed network of components without affecting the operation of the system. For example, the various components can be located in a switch such as a PBX and media server, gateway, in one or more communications devices, at one or more users' premises, or some combination thereof. Similarly, one or more functional portions of the system could be distributed between a telecommunications device(s) and an associated computing device.

[0109] Furthermore, it should be appreciated that the various links connecting the elements can be wired or wireless links, or any combination thereof, or any other known or later developed element(s) that is capable of supplying and/or communicating data to and from the connected elements. These wired or wireless links can also be secure links and may be capable of communicating encrypted information. Transmission media used as links, for example, can be any suitable carrier for electrical signals, including coaxial cables, copper wire and fiber optics, and may take the form of acoustic or light waves, such as those generated during radio-wave and infra-red data communications.

[0110] Also, while the flowcharts have been discussed and illustrated in relation to a particular sequence of events, it should be appreciated that changes, additions, and omissions to this sequence can occur without materially affecting the operation of the disclosed embodiments, configuration, and aspects.

[0111] A number of variations and modifications of the disclosure can be used. It would be possible to provide for some features of the disclosure without providing others.

[0112] For example in one alternative embodiment, the virtual keyboard 704 can be displayed within one screen 104 or 108 only, for example when the screens 104, 108 are in a landscape orientation.

[0113] In another alternative embodiment, in a landscape orientation, a first portion of the virtual keyboard 704 can be displayed by the first screen 104, while a second portion of the virtual keyboard 704 can be displayed by the second screen 108.

[0114] In yet another embodiment, the systems and methods of this disclosure can be implemented in conjunction with a special purpose computer, a programmed microprocessor or microcontroller and peripheral integrated circuit element(s), an ASIC or other integrated circuit, a digital signal processor, a hard-wired electronic or logic circuit such as discrete element circuit, a programmable logic device or gate array such as PLD, PLA, FPGA, PAL, special purpose computer, any comparable means, or the like. In general, any device(s) or means capable of implementing the methodology illustrated herein can be used to implement the various aspects of this disclosure. Exemplary hardware that can be used for the disclosed embodiments, configurations and aspects includes computers, handheld devices, telephones (e.g., cellular, Internet enabled, digital, analog, hybrids, and others), and other hardware known in the art. Some of these devices include processors (e.g., a single or multiple microprocessors), memory, nonvolatile storage, input devices, and output devices. Furthermore, alternative software implementations including, but not limited to, distributed processing or component/object distributed processing, parallel processing, or virtual machine processing can also be constructed to implement the methods described herein.

[0115] In yet another embodiment, the disclosed methods may be readily implemented in conjunction with software using object or object-oriented software development environments that provide portable source code that can be used on a variety of computer or workstation platforms. Alternatively, the disclosed system may be implemented partially or fully in hardware using standard logic circuits or VLSI design. Whether software or hardware is used to implement the systems in accordance with this disclosure is dependent on the speed and/or efficiency requirements of the system, the particular function, and the particular software or hardware systems or microprocessor or microcomputer systems being utilized.

[0116] In yet another embodiment, the disclosed methods may be partially implemented in software that can be stored on a storage medium, executed on programmed general-purpose

computer with the cooperation of a controller and memory, a special purpose computer, a microprocessor, or the like. In these instances, the systems and methods of this disclosure can be implemented as program embedded on personal computer such as an applet, JAVA® or CGI script, as a resource residing on a server or computer workstation, as a routine embedded in a dedicated measurement system, system component, or the like. The system can also be implemented by physically incorporating the system and/or method into a software and/or hardware system.

[0117] Although the present disclosure describes components and functions implemented in the aspects, embodiments, and/or configurations with reference to particular standards and protocols, the aspects, embodiments, and/or configurations are not limited to such standards and protocols. Other similar standards and protocols not mentioned herein are in existence and are considered to be included in the present disclosure. Moreover, the standards and protocols mentioned herein and other similar standards and protocols not mentioned herein are periodically superseded by faster or more effective equivalents having essentially the same functions. Such replacement standards and protocols having the same functions are considered equivalents included in the present disclosure.

[0118] The present disclosure, in various aspects, embodiments, and/or configurations, includes components, methods, processes, systems and/or apparatus substantially as depicted and described herein, including various aspects, embodiments, configurations embodiments, subcombinations, and/or subsets thereof. Those of skill in the art will understand how to make and use the disclosed aspects, embodiments, and/or configurations after understanding the present disclosure. The present disclosure, in various aspects, embodiments, and/or configurations, includes providing devices and processes in the absence of items not depicted and/or described herein or in various aspects, embodiments, and/or configurations hereof, including in the absence of such items as may have been used in previous devices or processes, e.g., for improving performance, achieving ease and\or reducing cost of implementation.

[0119] The foregoing discussion has been presented for purposes of illustration and description. The foregoing is not intended to limit the disclosure to the form or forms disclosed herein. In the foregoing Detailed Description for example, various features of the disclosure are grouped together in one or more aspects, embodiments, and/or configurations for the purpose of streamlining the disclosure. The features of the aspects, embodiments, and/or configurations of

the disclosure may be combined in alternate aspects, embodiments, and/or configurations other than those discussed above. This method of disclosure is not to be interpreted as reflecting an intention that the claims require more features than are expressly recited in each claim. Rather, as the following claims reflect, inventive aspects lie in less than all features of a single foregoing disclosed aspect, embodiment, and/or configuration. Thus, the following claims are hereby incorporated into this Detailed Description, with each claim standing on its own as a separate preferred embodiment of the disclosure.

[0120] Moreover, though the description has included description of one or more aspects, embodiments, and/or configurations and certain variations and modifications, other variations, combinations, and modifications are within the scope of the disclosure, e.g., as may be within the skill and knowledge of those in the art, after understanding the present disclosure. It is intended to obtain rights which include alternative aspects, embodiments, and/or configurations to the extent permitted, including alternate, interchangeable and/or equivalent structures, functions, ranges or steps to those claimed, whether or not such alternate, interchangeable and/or equivalent structures, functions, ranges or steps are disclosed herein, and without intending to publicly dedicate any patentable subject matter.

What is claimed is:

1. A method, comprising:

providing a device having first and second touch screens;

in a first operating mode, first presenting a virtual keyboard, wherein a first portion of the virtual keyboard is presented using a first portion of the first touch screen, and wherein a second portion of the virtual keyboard is presented using a first portion of the second touch screen.

2. The method of claim 1, further comprising:

in a second operating mode, second presenting a virtual keyboard, wherein all of the virtual keyboard is presented using one of the first touch screen and the second touch screen.

3. The method of claim 2, wherein the first operating mode is a dual portrait mode, and wherein the second operating mode is a dual landscape mode.

4. The method of claim 1, wherein in the first operating mode the method further comprises:

presenting using a second portion of the first touch screen first information.

5. The method of claim 4, further comprising:

presenting using a second portion of the second touch screen second information.

6. The method of claim 5, wherein in the second operating mode the method further comprises:

presenting using the second touch screen the first information and the second information.

7. The method of claim 3, wherein in the first operating mode the presentation of the virtual keyboard is modified as compared to the presentation of the virtual keyboard in the second operating mode, and wherein the modification includes aligning keys included in the virtual keyboard to accommodate a gap between a first group of the keys included in the virtual

keyboard that are presented by the first touch screen and a second group of keys included in the virtual keyboard that are presented by the second touch screen.

8. The method of claim 3, wherein the virtual keyboard has at least one dimension that is the same in the first operating mode as in the second operating mode.

9. The method of claim 2, wherein the first operating mode is entered when the device is in a first orientation, and wherein the second operating mode is entered when the device is in a second orientation.

10. The method of claim 1, wherein the virtual keyboard is a virtual alphanumeric keyboard.

11. A device, comprising:

a first screen, the first screen including a touch sensitive display area;

a second screen, the second screen including a touch sensitive display area;

memory;

a processor;

application programming stored in the memory and executed by the processor, wherein the application programming is operable to:

in a first operating mode, display a virtual keyboard using a portion of the touch sensitive display area of the first screen and using a portion of the touch sensitive display area of the second screen.

12. The device of claim 11, further comprising:

an accelerometer, wherein the accelerometer is operative to provide an indication of an orientation of the first and second touch screens to the application programming, and wherein the first operating mode is entered while first screen is in a portrait orientation and while the second screen is in a portrait orientation.

13. The device of claim 12, wherein the application programming is further operable to:

in a second operating mode, display the virtual keyboard using at least a portion of a first one of the touch sensitive display area of the first screen and the second screen, and without using any portion of a second one of the touch sensitive display area of the first screen and the second screen.

14. The device of claim 13, wherein the virtual keyboard is in a first configuration in the first operating mode, and wherein the virtual keyboard is in a second configuration in the second operating mode.

15. The device of claim 14, wherein the virtual keyboard presents a QWERTY keyboard to a user, and wherein in the first operating mode a line between the first screen and the

second screen extends between the T and the Y key, the G and the H key, and the V and the B key.

16. A computer readable medium having stored thereon computer-executable instructions, the computer executable instructions causing a processor to execute a method for providing a virtual keyboard, the computer-executable instructions comprising:

instructions to display a first portion of a virtual keyboard using a first portion of a first touch screen and to display a second portion of the virtual keyboard using a second portion of a second touch screen.

17. The computer readable medium of claim 16, further comprising:

instructions to determine an orientation of the first and second touch screens, wherein the instructions to display a first portion of a virtual keyboard using a first portion of a first touch screen and to display a second portion of the virtual keyboard using a second portion of a second touch screen are operative when the first and second touch screens are determined to be in a portrait mode.

18. The computer readable medium of claim 17, further comprising:

instructions to display a portion of first information using a second portion of the first touch screen and to display a portion of the first information using a second portion of the second touch screen while the first and second touch screens are determined to be in the portrait mode.

19. The computer readable medium of claim 18, further comprising:

instructions to display all of the virtual keyboard using a first one of the first touch screen and the second touch screen in response to a determination that the first and second touch screens are in a first landscape orientation.

20. The computer readable medium of claim 19, wherein at least some keys of the virtual keyboard with the first and second touch screens determined to be in the portrait orientation are shifted to accommodate a border between the first and second touch screens as compared to the keys of the virtual keyboard with the first and second touch screens determined to be in the landscape orientation.

ABSTRACT

Methods and devices for presenting a virtual keyboard are provided. More particularly, in connection with a multiple screen device, a virtual keyboard can be presented using portions of both of the screens. More particularly, with the screens of the device in a portrait orientation, the virtual keyboard can span the two screens, such that a first portion of the first screen and a first portion of the second screen operate cooperatively to present the virtual keyboard.

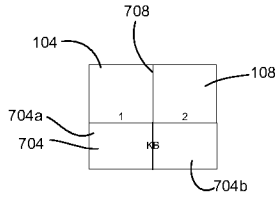


Fig. 7A

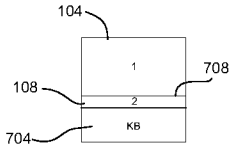


Fig. 7B

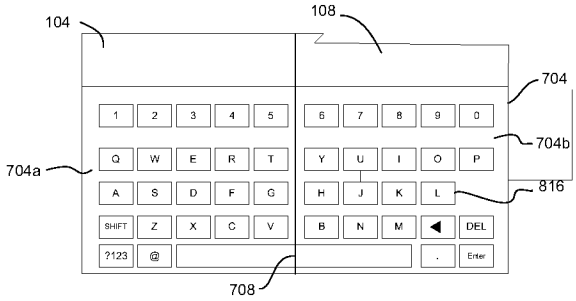


Fig. 8A

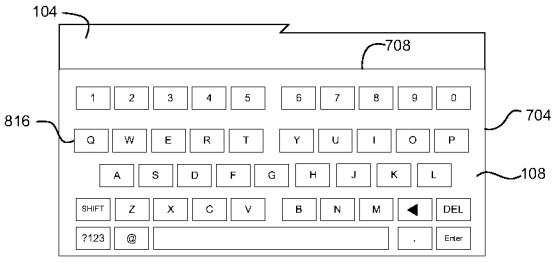


Fig. 8B

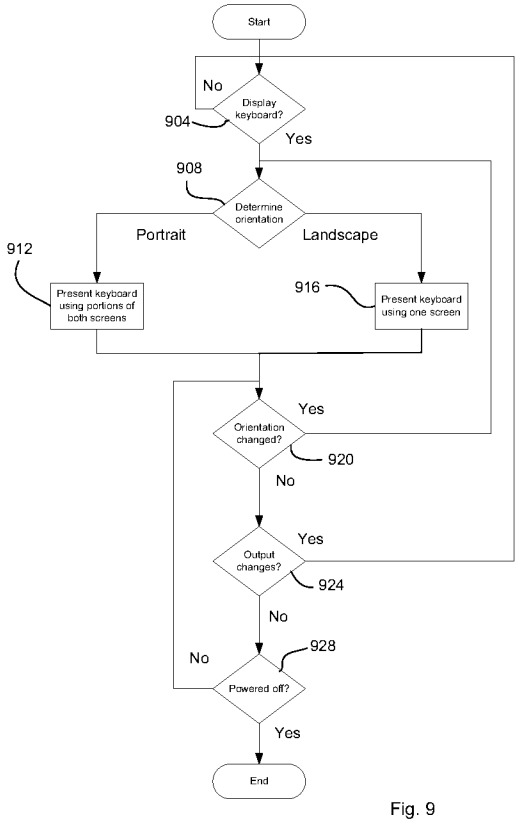


Fig. 9

KEYBOARD FILLING ONE SCREEN OR SPANNING MULTIPLE SCREENS OF A MULTIPLE SCREEN DEVICE

[0100] Fig. 7A depicts a device 100 with first 104 and second 108 screens. In this example, the screens 104, 108 are in portrait mode. A virtual keyboard 704 is presented by the screens 104, 108. In particular, a first portion 704a of the virtual keyboard 704 is presented by a first portion of the first screen 104, while a second portion 704b of the virtual keyboard 704 is presented by a first portion of the second screen 108. More particularly, the virtual keyboard 704 spans a line of separation 708 between the first and second screens. Moreover, the virtual keyboard 704 does not occupy all of the area of either the first screen 104 or the second screen 108. Accordingly, a second portion of the first screen 104 and a second portion of the second screen 108 remain available for the presentation of information other than the virtual keyboard 704 to a user.

[0101] Fig. 7B depicts a device 100 with first 104 and second 108 screens in a landscape mode. In this example, the first screen 104 is above and adjacent to the second screen 108. In addition, a virtual keyboard 704 is presented by the second screen 108. As shown, the virtual keyboard 704 can occupy all of the area of the second screen 108. In addition, the area occupied by the virtual keyboard 704 can be the same or about the same (*e.g.*, within ±10%) while the screens 104, 108 are in either portrait or landscape modes. In accordance with still other embodiments, the virtual keyboard 704 can be presented by the first screen 104 while in a landscape mode. For example, where the device 100 is held or positioned such that the second screen 108 is above the first screen 104, the virtual keyboard 704 can be presented by the first screen 104. In accordance with still further embodiments of the present invention, a virtual keyboard 704 can be presented using portions of the first 104 and second 108 screens, even when those screens are in a landscape mode.

[0102] Fig. 8A presents details of an exemplary virtual keyboard 704, with a first portion 704a presented using a first portion of a first screen 104 and with a second portion 704b presented using a first portion of a second screen 108 of a device 100 in accordance with embodiments of the present invention. As shown in this example, and without limitation, the virtual keyboard 704 can comprise a QWERTY keyboard. Because the virtual keyboard 704 is presented using two adjacent screens 104, 108, the line of separation 708 passes between a first portion 704a of the virtual keyboard 704 presented using the first screen 104, and a second portion 704b of the virtual keyboard 704 presented by the second screen 108. Moreover, because of the line of separation 708, the spacing of the keys 816 of the virtual keyboard 704 can be arranged such that the line of separation 708 falls between individual character keys 816. A space bar 820 can span across both the first 104 and second 108 screens.

[0103] Fig. 8B depicts a virtual keyboard 704 displayed by one of the screens 104 or 108 when the screens 104, 108 are in a landscape mode. In particular, in this example, the virtual keyboard 704 is displayed by the second screen 108, while the area of the first screen 104 remains available to present information other than the virtual keyboard 704 to a user. Moreover, the virtual keyboard occupies all of the second screen 108. As shown, in a dual landscape mode the virtual keyboard 704 can be presented in the screen 104, 108 that is below the other screen 104, 108.

[0104] Comparing the virtual keyboard 704 presented with portions of the screens 104, 108 in a portrait mode, such that the virtual keyboard 704 spans the two screens 104, 108, to the mode of operation in which the virtual keyboard 704 is presented by a single screen 104 or 108 in a landscape mode highlights changes that can be made to accommodate the line of separation 708. In particular, in the embodiment illustrated in Fig. 8A, the spacing of the keys 816 has been altered so that the smaller keys (*e.g.*, the character keys) lie on one side or the other of the line of separation 708. In contrast, in the embodiment illustrated in Fig. 8B, the keys 816 of the virtual keyboard 704 have a more conventional, interleaved type spacing.

[0105] Fig. 9 depicts aspects of a method for presenting a virtual keyboard 704 in accordance with embodiments of the present invention. Initially, at step 904, a determination is made as to whether a virtual keyboard is to be displayed. Until such a determination is made, the process may idle at step 904. After a determination to display a virtual keyboard 704 has been made, a determination can next be made as to the orientation of the device 100 (step 908). If the device 100 is in a portrait mode, the virtual keyboard 704 is presented using portions of both the first 104 and second 108 screens of the multiple screen device 100 (step 912). Accordingly, with the screens 104, 108 in the portrait mode, the virtual keyboard 704 spans portions of both screens 104, 108. If the device 100 screens 104, 108 are determined to be in a landscape orientation, the virtual keyboard 704 can be presented using one of the screens 104 or 108 (step 916). For example, the virtual keyboard 704 can be presented by the screen 104 or 108 that is below the other screen 104 or 108. After presenting the virtual keyboard 704 in the mode determined to be

correct, a determination can be made as to whether the orientation of the device has changed (step 920). If the orientation has changed, the process can return to step 908. If the orientation has not changed, a determination can next be made as to whether the content of the displayed output has changed (step 924). A change to the output can include the display of characters entered by a user through the virtual keyboard 704. In accordance with other embodiments, a change to the output can include a change to the information displayed through operation of application programming and/or the receipt of information at the device 100. A change in output can also include an acknowledgement of commands received from a user or from a connected device, or that are generated through operation of application programming, and the effect or processing of the command. After determining that the output provided by the displays 104, 108 has changed, the process can return to step 904, to determine whether, in view of the change or other information, the virtual keyboard 704 should continue to be displayed.

[0106] At step 928, a determination can be made as to whether operation of the device should continue, or whether the device has powered down. If operation is to continue, the process can return to step 904. If the device has been powered down, the process can end. Although steps in the process have been shown and described in a particular order, it should be appreciated that other orderings are possible and incorporated herein. For example, the process can be discontinued at any point in the process if the device 100 is powered off.

[0107] In accordance with embodiments of the present invention, the method disclosed herein can be performed by the execution of application programming or other software components stored in memory 208, 508 by a processor 204, 504. For instance, the operating system 516, a component or components of the framework 520, and/or a component or components of the multi-display management module 524 can include or provide functionality to receive and act on input received from a user in connection with a virtual keyboard 704. Moreover, such input can include input comprising user interface interactions by a user that are managed by the input management module 536, or other components. In addition, although certain embodiments have been described in connection with operation on a device having first 104 and second 108 screens, the invention is not limited to operation on such a device. For example, a device 100 can have more than two screens, and/or the screen can comprise virtual screens or windows.

[0108] The exemplary systems and methods of this disclosure have been described in relation to the presentation of a virtual keyboard 704 that in a dual landscape mode occupies all or

substantially all of the bottom screen 104 or 108 of a device 100 with multiple screens 104, 108. However, to avoid unnecessarily obscuring the present disclosure, the preceding description omits a number of known structures and devices. This omission is not to be construed as a limitation of the scopes of the claims. Specific details are set forth to provide an understanding of the present disclosure. It should however be appreciated that the present disclosure may be practiced in a variety of ways beyond the specific detail set forth herein.

[0109] Furthermore, while the exemplary aspects, embodiments, and/or configurations illustrated herein show the various components of the system collocated, certain components of the system can be located remotely, at distant portions of a distributed network, such as a LAN and/or the Internet, or within a dedicated system. Thus, it should be appreciated, that the components of the system can be combined in to one or more devices, such as a computer, laptop, netbook, tablet computer, smart phone, mobile device, etc., or collocated on a particular node of a distributed network, such as an analog and/or digital telecommunications network, a packet-switch network, or a circuit-switched network. It will be appreciated from the preceding description, and for reasons of computational efficiency, that the components of the system can be arranged at any location within a distributed network of components without affecting the operation of the system. For example, the various components can be located in a switch such as a PBX and media server, gateway, in one or more communications devices, at one or more users' premises, or some combination thereof. Similarly, one or more functional portions of the system could be distributed between a telecommunications device(s) and an associated computing device.

[0110] Furthermore, it should be appreciated that the various links connecting the elements can be wired or wireless links, or any combination thereof, or any other known or later developed element(s) that is capable of supplying and/or communicating data to and from the connected elements. These wired or wireless links can also be secure links and may be capable of communicating encrypted information. Transmission media used as links, for example, can be any suitable carrier for electrical signals, including coaxial cables, copper wire and fiber optics, and may take the form of acoustic or light waves, such as those generated during radio-wave and infra-red data communications.

[0111] Also, while the flowcharts have been discussed and illustrated in relation to a particular sequence of events, it should be appreciated that changes, additions, and omissions to this

sequence can occur without materially affecting the operation of the disclosed embodiments, configuration, and aspects.

[0112] A number of variations and modifications of the disclosure can be used. It would be possible to provide for some features of the disclosure without providing others.

[0113] For example in one alternative embodiment, the virtual keyboard 704 can be displayed within a portion of one screen 104 or 108 when the screens 104, 108 are in a landscape orientation.

[0114] In another alternative embodiment, in a landscape orientation, a first portion of the virtual keyboard 704 can be displayed by the first screen 104, while a second portion of the virtual keyboard 704 can be displayed by the second screen 108.

[0115] In yet another embodiment, the systems and methods of this disclosure can be implemented in conjunction with a special purpose computer, a programmed microprocessor or microcontroller and peripheral integrated circuit element(s), an ASIC or other integrated circuit, a digital signal processor, a hard-wired electronic or logic circuit such as discrete element circuit, a programmable logic device or gate array such as PLD, PLA, FPGA, PAL, special purpose computer, any comparable means, or the like. In general, any device(s) or means capable of implementing the methodology illustrated herein can be used to implement the various aspects of this disclosure. Exemplary hardware that can be used for the disclosed embodiments, configurations and aspects includes computers, handheld devices, telephones (e.g., cellular, Internet enabled, digital, analog, hybrids, and others), and other hardware known in the art. Some of these devices include processors (e.g., a single or multiple microprocessors), memory, nonvolatile storage, input devices, and output devices. Furthermore, alternative software implementations including, but not limited to, distributed processing or component/object distributed processing, parallel processing, or virtual machine processing can also be constructed to implement the methods described herein.

[0116] In yet another embodiment, the disclosed methods may be readily implemented in conjunction with software using object or object-oriented software development environments that provide portable source code that can be used on a variety of computer or workstation platforms. Alternatively, the disclosed system may be implemented partially or fully in hardware using standard logic circuits or VLSI design. Whether software or hardware is used to implement the systems in accordance with this disclosure is dependent on the speed and/or

efficiency requirements of the system, the particular function, and the particular software or hardware systems or microprocessor or microcomputer systems being utilized.

[0117] In yet another embodiment, the disclosed methods may be partially implemented in software that can be stored on a storage medium, executed on programmed general-purpose computer with the cooperation of a controller and memory, a special purpose computer, a microprocessor, or the like. In these instances, the systems and methods of this disclosure can be implemented as program embedded on personal computer such as an applet, JAVA® or CGI script, as a resource residing on a server or computer workstation, as a routine embedded in a dedicated measurement system, system component, or the like. The system can also be implemented by physically incorporating the system and/or method into a software and/or hardware system.

[0118] Although the present disclosure describes components and functions implemented in the aspects, embodiments, and/or configurations with reference to particular standards and protocols, the aspects, embodiments, and/or configurations are not limited to such standards and protocols. Other similar standards and protocols not mentioned herein are in existence and are considered to be included in the present disclosure. Moreover, the standards and protocols mentioned herein and other similar standards and protocols not mentioned herein are periodically superseded by faster or more effective equivalents having essentially the same functions. Such replacement standards and protocols having the same functions are considered equivalents included in the present disclosure.

[0119] The present disclosure, in various aspects, embodiments, and/or configurations, includes components, methods, processes, systems and/or apparatus substantially as depicted and described herein, including various aspects, embodiments, configurations embodiments, subcombinations, and/or subsets thereof. Those of skill in the art will understand how to make and use the disclosed aspects, embodiments, and/or configurations after understanding the present disclosure. The present disclosure, in various aspects, embodiments, and/or configurations, includes providing devices and processes in the absence of items not depicted and/or described herein or in various aspects, embodiments, and/or configurations hereof, including in the absence of such items as may have been used in previous devices or processes, e.g., for improving performance, achieving ease and\or reducing cost of implementation.

[0120] The foregoing discussion has been presented for purposes of illustration and description. The foregoing is not intended to limit the disclosure to the form or forms disclosed herein. In the foregoing Detailed Description for example, various features of the disclosure are grouped together in one or more aspects, embodiments, and/or configurations for the purpose of streamlining the disclosure. The features of the aspects, embodiments, and/or configurations of the disclosure may be combined in alternate aspects, embodiments, and/or configurations other than those discussed above. This method of disclosure is not to be interpreted as reflecting an intention that the claims require more features than are expressly recited in each claim. Rather, as the following claims reflect, inventive aspects lie in less than all features of a single foregoing disclosed aspect, embodiment, and/or configuration. Thus, the following claims are hereby incorporated into this Detailed Description, with each claim standing on its own as a separate preferred embodiment of the disclosure.

[0121] Moreover, though the description has included description of one or more aspects, embodiments, and/or configurations and certain variations and modifications, other variations, combinations, and modifications are within the scope of the disclosure, e.g., as may be within the skill and knowledge of those in the art, after understanding the present disclosure. It is intended to obtain rights which include alternative aspects, embodiments, and/or configurations to the extent permitted, including alternate, interchangeable and/or equivalent structures, functions, ranges or steps to those claimed, whether or not such alternate, interchangeable and/or equivalent structures, functions, ranges or steps are disclosed herein, and without intending to publicly dedicate any patentable subject matter.

What is claimed is:

1. A method, comprising:

providing a device having first and second touch screens;

selecting from first and second operating modes;

in the first operating mode, first presenting a virtual keyboard, wherein a first portion of the virtual keyboard is presented using a first portion of the first touch screen, and wherein a second portion of the virtual keyboard is presented using a first portion of the second touch screen;

in the second operating mode, second presenting a virtual keyboard, wherein all of the virtual keyboard is presented using one of the first touch screen and the second touch screen.

2. The method of claim 1, wherein selecting from first and second operating modes includes determining an orientation of the first and second touch screens, wherein the first operating mode is selected in response to determining that the first and second touch screens are in a portrait mode, and wherein the second operating mode is selected in response to determining that the first and second touch screens are in a landscape mode.

3. The method of claim 2, wherein the portrait mode is a dual portrait mode, and wherein the landscape mode is a dual landscape mode.

4. The method of claim 3, wherein in the first operating mode the method further comprises:

presenting using a second portion of the first touch screen first information.

5. The method of claim 4, further comprising:

presenting using a second portion of the second touch screen second information.

6. The method of claim 5, wherein in the second operating mode the method further comprises:

presenting using the second touch screen the first information and the second information.

7. The method of claim 1, wherein in the first operating mode the presentation of the virtual keyboard is modified as compared to the presentation of the virtual keyboard in the second operating mode, and wherein the modification includes aligning keys included in the virtual keyboard to accommodate a gap between a first group of the keys included in the virtual keyboard that are presented by the first touch screen and a second group of keys included in the virtual keyboard that are presented by the second touch screen.

8. The method of claim 1, wherein the virtual keyboard has at least one dimension that is the same in the first operating mode as in the second operating mode.

9. The method of claim 1, wherein the first operating mode is entered when the device is in a first orientation, and wherein the second operating mode is entered when the device is in a second orientation.

10. The method of claim 1, wherein the virtual keyboard is a virtual alphanumeric keyboard.

11. A device, comprising:

a first screen, the first screen including a touch sensitive display area;

a second screen, the second screen including a touch sensitive display area;

memory;

a processor;

application programming stored in the memory and executed by the processor, wherein the application programming is operable to:

in a first operating mode, display a virtual keyboard using a portion of the touch sensitive display area of the first screen and using a portion of the touch sensitive display area of the second screen;

in a second operating mode, display the virtual keyboard using at least a portion of a first one of the touch sensitive display area of the first screen and the second screen, and without using any portion of a second one of the touch sensitive display area of the first screen and the second screen.

12. The device of claim 11, wherein in the second operating mode all of the first one of the first screen and second screen is used to display the virtual keyboard.

13. The device of claim 11, further comprising:

an accelerometer, wherein the accelerometer is operative to provide an indication of an orientation of the first and second touch screens to the application programming, and wherein the first operating mode is entered while first screen is in a portrait orientation and while the second screen is in a portrait orientation.

14. The device of claim 13, wherein the virtual keyboard is in a first configuration in the first operating mode, and wherein the virtual keyboard is in a second configuration in the second operating mode.

15. The device of claim 14, wherein the virtual keyboard presents a QWERTY keyboard to a user, and wherein in the first operating mode a line between the first screen and the

second screen extends between the T and the Y key, the G and the H key, and the V and the B key.

16. A computer readable medium having stored thereon computer-executable instructions, the computer executable instructions causing a processor to execute a method for providing a virtual keyboard, the computer-executable instructions comprising:

instructions to display all of the virtual keyboard using a first one of a first touch screen and a second touch screen.

17. The computer readable medium of claim 16, further comprising:

instructions to determine an orientation of the first and second touch screens, wherein the instructions to display all of the virtual keyboard using a first one of the first touch screen and the second touch screen are operative when the first and second touch screens are determined to be in a landscape mode.

18. The computer readable medium of claim 17, further comprising:

instructions to display a portion of first information using a second portion of the first touch screen and to display a portion of the first information using a second portion of the second touch screen while the first and second touch screens are determined to be in the portrait mode.

19. The computer readable medium of claim 18, further comprising:

instructions to display a first portion of a virtual keyboard using a first portion of a first touch screen and to display a second portion of the virtual keyboard using a second portion of a second touch screen.

20. The computer readable medium of claim 19, wherein at least some keys of the virtual keyboard with the first and second touch screens determined to be in the portrait orientation are shifted to accommodate a border between the first and second touch screens as compared to the keys of the virtual keyboard with the first and second touch screens determined to be in the landscape orientation.

ABSTRACT

Methods and devices for presenting a virtual keyboard are provided. More particularly, in connection with a multiple screen device, a virtual keyboard can be presented in a first mode using all of one of the screens. In a second mode, the virtual keyboard can be presented using portions of both of the screens. More particularly, with the screens of the device in a landscape orientation, one screen can be devoted to present the virtual keyboard while the other screen remains available to present other information. In a portrait orientation, the virtual keyboard can span the two screens, such that a first portion of the first screen and a first portion of the second screen operate cooperatively to present the virtual keyboard.

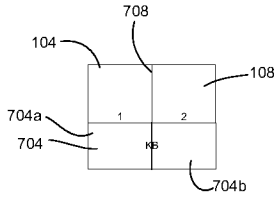


Fig. 7A

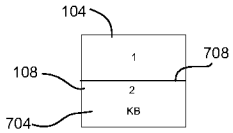


Fig. 7B

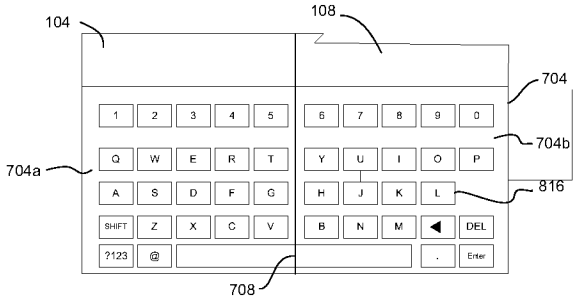


Fig. 8A

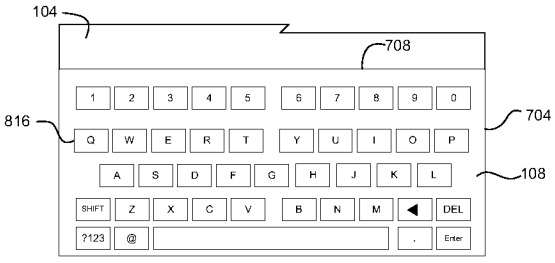


Fig. 8B

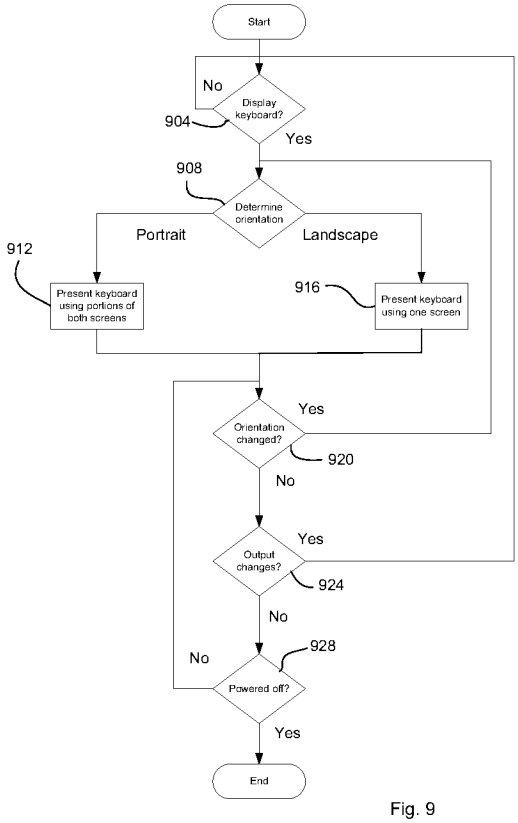


Fig. 9

FOCUS CHANGE DISMISSES VIRTUAL KEYBOARD ON A MULTIPLE SCREEN DEVICE

[0100] Fig. 7A depicts a multiple screen device 100 with first 104 and second 108 screens. In this example, the screens 104, 108 are each in a landscape mode or orientation. In the state depicted in Fig. 7A, the first screen 104 displays a first window or set of information, for example a window comprising a first application 708. The second screen 108 displays a second set of information, for example a window comprising a second application 712.

[0101] In Fig. 7B, the multiple screen device 100 is shown after an input mode has been entered, with the determined focus of the user on the first set of information or window 708. In this mode, a virtual keyboard 704 is displayed by the second screen 108. Moreover, because the focus of the user is determined to be on the first window 708, the first screen 104 displays the first window 708.

[0102] In Fig. 8A, the device 100 is shown in an input mode, with a virtual keyboard 704 again displayed by the second screen 108. Moreover, the mode depicted in Fig. 8A can be entered from an initial state as depicted in Fig. 7A. In the state depicted in Fig. 8A, the first screen 104 displays the second window 712. Accordingly, this state can be entered from a starting point illustrated in Fig. 7A, where an input mode is entered, and where the focus of the user is determined to be on the second set of information or window 712. Accordingly, when the input mode is entered, the first screen 104 discontinues a display of the first window 708, and instead displays the information comprising the second window 712, that was formerly presented by the second screen 108. The virtual keyboard 704 can therefore be presented in its expected place, in the lower screen (here the second screen 108), while the window having focus (here the second window 712) can continue to be presented to the user by moving that second window 712 to the first screen 104.

[0103] Fig. 8B depicts the device 100 of Fig. 8A, after the virtual keyboard 704 has been dismissed. In this example, the second screen 108 displays a desktop 716. Accordingly, the action by which the second window 712 was pushed to the first screen 104 is not undone when the keyboard 704 is dismissed.

[0104] With reference now to Fig. 9, aspects of a method for controlling a display or presentation of a virtual keyboard 704 in accordance with embodiments of the present invention are depicted. Initially, at step 904, a determination is made as to whether a virtual keyboard 704 is to be displayed. Until such a determination is made, the process may idle at step 904. In general, a determination to display a virtual keyboard 704 can be made in response to entering an input mode, either explicitly through a command received from the user, or implicitly, through operation of application programming independently of an explicit command from the user, or implicitly in response to user input. After a determination is made to display a virtual keyboard 704, a window having the focus of the user is identified (step 908). A determination can then be made as to whether the window having the focus of the user is on the lower screen of the two screens 104, 108 when those screens are in a dual landscape format or mode (step 912). If the focus of the user is determined to be on a window (*e.g.*, window 712) presented by the lower screen (*e.g.*, the second screen 108), that window is pushed to the upper screen (*e.g.*, the first screen 104), and the virtual keyboard is displayed by the lower screen (*e.g.*, second screen 108) (step 916). If the window having focus is determined to not be on the lower screen, the virtual keyboard is displayed on the lower screen in place of the window (*e.g.*, second window 712), desktop, or other information formerly displayed by the lower screen (step 920). Moreover, where the focus is not determined to be on the information in the lower screen, the window with focus (*e.g.*, the first window 708) continues to be displayed by, for example, the first screen 104. Afrer displaying the virtual keyboard 704 at step 916 or 920, a determination can be made as to whether user input has been received in the area occupied by the virtual keyboard 704 (step 924). If input is received, for example a user input comprising a key press associated with a key presented by the virtual keyboard 704, that input can be delivered to the application associated with the window that presently has the focus of the user, and can be acted on by that application or associated programming (step 928).

[0105] Afrer receiving and acting on received input at step 928, or afrer determining that input has not been received by the virtual keyboard at step 924, a determination can be made as to whether the keyboard should be dismissed (step 932). In accordance with embodiments of the present invention, a virtual keyboard 704 should be dismissed in response to user input explicitly dismissing the keyboard 704, or input comprising a window management operation. For example, user input entered within a gesture capture region 120, 124 within a configurable area 112, 116, or within any portion of a touch sensitive display 110, 114 that is outside of the area of the virtual keyboard 704 results in the virtual keyboard 704 being dismissed. In response to a determination that the virtual keyboard 704 should be dismissed, a determination is made as to

whether the presentation of the virtual keyboard 704 was associated with a push of displayed information from one of the screens 104, 108 to the other 104 or 108 (step 936). If information, such as a window 708 or 712 was pushed to a different screen 104 or 108, that push is not undone. Instead, the dismissal of the keyboard results in the window 708, 712 that was pushed to the top screen 104 or 108 remains in the top screen 104 or 108, and desktop information or a second portion of the information displayed by the window presented by the top screen 104 or 108 is presented by the bottom screen 104 or 108 (step 940). Alternatively, if the presentation of the virtual keyboard 704 did not result in a window 708 or 712 being pushed to the top screen 104 or 108, then the application or window 708 or 712 that had been displayed by the lower screen 104 or 108 is again presented by that lower screen 104 or 108 when the keyboard is dismissed (step 944).

[0106] At step 948, a determination can be made as to whether the device 100 has been powered off. If the device 100 has not been powered off, the process can return to step 904. Alternatively, the process may end.

[0107] Although examples provided herein discuss the first screen 104 as the top screen and the second screen 108 as the bottom screen when the device 100 is in a dual landscape mode, such is not necessarily the case. In particular, as can be appreciated afrer consideration of the present disclosure, the device 100 can be in a dual landscape mode with the second screen 108 operating as the top screen and the first screen 104 operating as the bottom screen.

[0108] In accordance with embodiments of the present invention, the methods disclosed herein can be performed by the execution of application programming stored in memory 208, 508 and by a processor 204, 504. For instance, a windows management module or class 532 can include functionality to receive and act on input received from the user. Moreover, various application programming can operate to determine when a virtual keyboard 704 should be presented. Moreover, such input can include window management input, and can include the execution of steps to discontinue the presentation of a virtual keyboard 704 in response to such input. A windows management module or class 532 or other module provided as part of a framework 520 can provide the virtual keyboard 704 and can receive input provided by a user in connection with the virtual keyboard 704. In addition, although certain embodiments have been described in connection with operation on a device having first 104 and second 108 screens, the invention is

not limited to operation on such a device. For example, a device 100 can have more than two screens, and/or the screen can comprise virtual screens or windows.

[0109] The exemplary systems and methods of this disclosure have been described in relation to controlling the display of a virtual keyboard 704 on a device 100 having multiple touch sensitive screens 104, 108. However, to avoid unnecessarily obscuring the present disclosure, the preceding description omits a number of known structures and devices. This omission is not to be construed as a limitation of the scopes of the claims. Specific details are set forth to provide an understanding of the present disclosure. It should however be appreciated that the present disclosure may be practiced in a variety of ways beyond the specific detail set forth herein.

[0110] Furthermore, while the exemplary aspects, embodiments, and/or configurations illustrated herein show the various components of the system collocated, certain components of the system can be located remotely, at distant portions of a distributed network, such as a LAN and/or the Internet, or within a dedicated system. Thus, it should be appreciated, that the components of the system can be combined in to one or more devices, such as a computer, laptop, netbook, tablet computer, smart phone, mobile device, etc., or collocated on a particular node of a distributed network, such as an analog and/or digital telecommunications network, a packet-switch network, or a circuit-switched network. It will be appreciated from the preceding description, and for reasons of computational efficiency, that the components of the system can be arranged at any location within a distributed network of components without affecting the operation of the system. For example, the various components can be located in a switch such as a PBX and media server, gateway, in one or more communications devices, at one or more users' premises, or some combination thereof. Similarly, one or more functional portions of the system could be distributed between a telecommunications device(s) and an associated computing device.

[0111] Furthermore, it should be appreciated that the various links connecting the elements can be wired or wireless links, or any combination thereof, or any other known or later developed element(s) that is capable of supplying and/or communicating data to and from the connected elements. These wired or wireless links can also be secure links and may be capable of communicating encrypted information. Transmission media used as links, for example, can be any suitable carrier for electrical signals, including coaxial cables, copper wire and fiber optics,

and may take the form of acoustic or light waves, such as those generated during radio-wave and infra-red data communications.

[0112] Also, while the flowcharts have been discussed and illustrated in relation to a particular sequence of events, it should be appreciated that changes, additions, and omissions to this sequence can occur without materially affecting the operation of the disclosed embodiments, configuration, and aspects.

[0113] A number of variations and modifications of the disclosure can be used. It would be possible to provide for some features of the disclosure without providing others.

[0114] For example in one alternative embodiment, a window 708, 712 and/or a virtual keyboard 704 can be presented by portions of multiple screens 104, 108, for example where the screens 104, 108 of the device 100 are positioned in a dual portrait mode.

[0115] In another alternative embodiment, different virtual keyboards 704 can be presented under different conditions. For example, in a text entry mode, a QWERTY keyboard consisting of letters and control keys can be presented, while in a numeric entry mode a keypad including numbers and arithmetic operations can be presented.

[0116] In yet another embodiment, the systems and methods of this disclosure can be implemented in conjunction with a special purpose computer, a programmed microprocessor or microcontroller and peripheral integrated circuit element(s), an ASIC or other integrated circuit, a digital signal processor, a hard-wired electronic or logic circuit such as discrete element circuit, a programmable logic device or gate array such as PLD, PLA, FPGA, PAL, special purpose computer, any comparable means, or the like. In general, any device(s) or means capable of implementing the methodology illustrated herein can be used to implement the various aspects of this disclosure. Exemplary hardware that can be used for the disclosed embodiments, configurations and aspects includes computers, handheld devices, telephones (e.g., cellular, Internet enabled, digital, analog, hybrids, and others), and other hardware known in the art. Some of these devices include processors (e.g., a single or multiple microprocessors), memory, nonvolatile storage, input devices, and output devices. Furthermore, alternative software implementations including, but not limited to, distributed processing or component/object distributed processing, parallel processing, or virtual machine processing can also be constructed to implement the methods described herein.

[0117] In yet another embodiment, the disclosed methods may be readily implemented in conjunction with software using object or object-oriented software development environments that provide portable source code that can be used on a variety of computer or workstation platforms. Alternatively, the disclosed system may be implemented partially or fully in hardware using standard logic circuits or VLSI design. Whether software or hardware is used to implement the systems in accordance with this disclosure is dependent on the speed and/or efficiency requirements of the system, the particular function, and the particular software or hardware systems or microprocessor or microcomputer systems being utilized.

[0118] In yet another embodiment, the disclosed methods may be partially implemented in software that can be stored on a storage medium, executed on programmed general-purpose computer with the cooperation of a controller and memory, a special purpose computer, a microprocessor, or the like. In these instances, the systems and methods of this disclosure can be implemented as program embedded on personal computer such as an applet, JAVA® or CGI script, as a resource residing on a server or computer workstation, as a routine embedded in a dedicated measurement system, system component, or the like. The system can also be implemented by physically incorporating the system and/or method into a software and/or hardware system.

[0119] Although the present disclosure describes components and functions implemented in the aspects, embodiments, and/or configurations with reference to particular standards and protocols, the aspects, embodiments, and/or configurations are not limited to such standards and protocols. Other similar standards and protocols not mentioned herein are in existence and are considered to be included in the present disclosure. Moreover, the standards and protocols mentioned herein and other similar standards and protocols not mentioned herein are periodically superseded by faster or more effective equivalents having essentially the same functions. Such replacement standards and protocols having the same functions are considered equivalents included in the present disclosure.

[0120] The present disclosure, in various aspects, embodiments, and/or configurations, includes components, methods, processes, systems and/or apparatus substantially as depicted and described herein, including various aspects, embodiments, configurations embodiments, subcombinations, and/or subsets thereof. Those of skill in the art will understand how to make and use the disclosed aspects, embodiments, and/or configurations after understanding the

present disclosure. The present disclosure, in various aspects, embodiments, and/or configurations, includes providing devices and processes in the absence of items not depicted and/or described herein or in various aspects, embodiments, and/or configurations hereof, including in the absence of such items as may have been used in previous devices or processes, e.g., for improving performance, achieving ease and\or reducing cost of implementation.

[0121] The foregoing discussion has been presented for purposes of illustration and description. The foregoing is not intended to limit the disclosure to the form or forms disclosed herein. In the foregoing Detailed Description for example, various features of the disclosure are grouped together in one or more aspects, embodiments, and/or configurations for the purpose of streamlining the disclosure. The features of the aspects, embodiments, and/or configurations of the disclosure may be combined in alternate aspects, embodiments, and/or configurations other than those discussed above. This method of disclosure is not to be interpreted as reflecting an intention that the claims require more features than are expressly recited in each claim. Rather, as the following claims reflect, inventive aspects lie in less than all features of a single foregoing disclosed aspect, embodiment, and/or configuration. Thus, the following claims are hereby incorporated into this Detailed Description, with each claim standing on its own as a separate preferred embodiment of the disclosure.

[0122] Moreover, though the description has included description of one or more aspects, embodiments, and/or configurations and certain variations and modifications, other variations, combinations, and modifications are within the scope of the disclosure, e.g., as may be within the skill and knowledge of those in the art, after understanding the present disclosure. It is intended to obtain rights which include alternative aspects, embodiments, and/or configurations to the extent permitted, including alternate, interchangeable and/or equivalent structures, functions, ranges or steps to those claimed, whether or not such alternate, interchangeable and/or equivalent structures, functions, ranges or steps are disclosed herein, and without intending to publicly dedicate any patentable subject matter.

What is claimed is:

1. A method for controlling a display of a virtual keyboard, comprising:

determining a focus of a user of a multiple screen device, wherein the multiple screen device has at least first and second screens, and wherein the focus of the user is determined to be on a window displayed by one of the first and second screens;

presenting the window having the determined focus by the first screen of the multiple screen device;

presenting the virtual keyboard by the second screen of the multiple screen device.

2. The method of claim 1, further comprising:

determining that a focus related to information presented by the first and second screens has changed;

in response to determining that the focus has changed, discontinuing the presentation of the virtual keyboard by the second screen of the device.

3. The method of claim 2, wherein the first input is received in a gesture capture region, and wherein the gesture capture region is included as part of at least one of the first and second screens of the device.

4. The method of claim 1, further comprising:

determining that a user input mode has been entered, wherein the virtual keyboard is presented in response to determining that the user input mode has been entered.

5. The method of claim 1, wherein a first window is presented by the first screen of the multiple screen device, wherein the focus of the user is determined to be on a second window presented by the second screen of the multiple screen device, wherein presenting the virtual keyboard by the second display includes presenting the second window window having the focus of the user by the first screen of the multiple screen device.

6. The method of claim 1, wherein the focus of the user is determined to be on a first window presented by the first screen of the multiple screen device, wherein presenting the virtual keyboard by the second screen includes discontinuing presenting a second window by the second screen of the multiple screen device.

7. The method of claim 6, further comprising:

determining that a focus related to information presented by the first and second screens has changed;

in response to determining that the focus has changed, discontinuing presenting the virtual keyboard by the second screen of the multiple screen device.

8. The method of claim 7, further comprising:

after discontinuing presenting the virtual keyboard by the second screen of the multiple screen device, presenting the second window by the second screen of the multiple screen device.

9. The method of claim 7, further comprising:

after discontinuing presenting the virtual keyboard by the second screen of the multiple screen device, presenting a third window by the second screen of the multiple screen device.

10. The method of claim 9, wherein the third window is one of an extension of the first window and a desktop.

11. A device, comprising:

a first screen, wherein the first screen includes a first touch sensitive display;

a second screen, wherein the second screen includes a second touch sensitive display;

memory;

a processor;

application programming stored in the memory and executed by the processor, wherein the application programming is operable to:

identify a window having a user focus;

display the window having the user focus on the first touch sensitive display;

display a virtual keyboard on the second touch sensitive display;

detect a change in the user focus;

in response to detecting the change in the user focus, discontinue the display of the virtual keyboard by the second touch sensitive display.

12. The device of claim 11, wherein the change in user focus is detected by a touch input received from the user entered on one of the first and second touch sensitive displays.

13. The device of claim 11, further comprising:

at least a first gesture capture region, wherein the change in user focus is detected by a touch input received from a user entered on the at least a first gesture capture region.

14. The device of claim 11, wherein the window having the user focus is identified by a touch input received ftom the user on one of the first and second touch sensitive displays.

15. The device of claim 14, wherein the touch input is received on the second touch sensitive display, wherein the window displayed by the touch sensitive display is displayed by the first touch screen display after receiving the touch input received on the second touch sensitive display, wherein the virtual keyboard is displayed on the second touch sensitive display after the touch input is received on the second touch sensitive display.

16. A computer readable medium having stored thereon computer-executable instructions, the computer executable instructions causing a processor to execute a method for controlling a display of a virtual keyboard, the computer-executable instructions comprising:

instructions to determine a focus of a user of a multiple screen device;

instructions to present a window having the focus of the user on a first screen of the multiple screen device;

instructions to present the virtual keyboard on a second screen of the multiple screen device.

17. The computer readable medium of claim 16, the computer executable instructions further comprising:

in response to determining that the focus of the user is on a window presented by the second screen of the multiple screen device, instructions to display the window having the focus of the user by the first screen of the multiple screen device.

18. The computer readable medium of claim 16, the computer executable instructions further comprising:

instructions to discontinue presenting the virtual keyboard in response to detecting a change in the focus of the user.

19. The computer readable medium of claim 16, wherein the virtual keyboard is presented on the second screen of the multiple screen device in place of a window that does not have the focus of the user.

20. The computer readable medium of claim 16, wherein the multiple screen device includes the computer readable medium.

ABSTRACT

Methods and devices for providing a virtual keyboard in connection with a multiple screen device are provided. More particularly, information displayed on the screen of a multiple screen device having a current focus of the user is identified, and is presented by a top screen. The virtual keyboard is presented by the bottom screen. The virtual keyboard can be dismissed in response to detecting a change in the focus of the user.

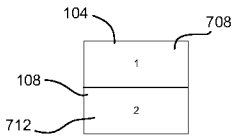


Fig. 7A

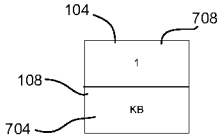


Fig. 7B

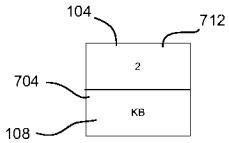


Fig. 8A

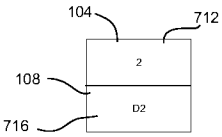


Fig. 8B

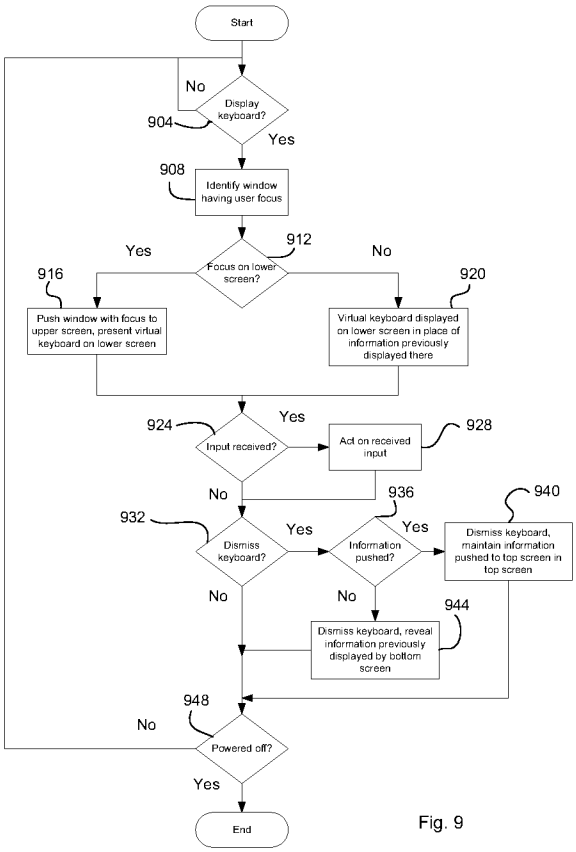


Fig. 9

KEYBOARD FILLS BOTTOM SCREEN ON ROTATION OF A MULTIPLE SCREEN DEVICE

[0100] Fig. 7A depicts a device 100 with first 104 and second 108 screens. In this example, the device 100 is depicted in a first mode or orientation in which the screens 104, 108 are in a dual portrait mode. A virtual keyboard 704 is presented by the screens 104, 108. In particular, a first portion 704a of the virtual keyboard 704 is presented by a first portion of the first screen 104, while a second portion 704b of the virtual keyboard 704 is presented by a first portion of the second screen 108. More particularly, the virtual keyboard 704 spans a line of separation 708 between the first and second screens. Moreover, the virtual keyboard 704 does not occupy all of the area of either the first screen 104 or the second screen 108. Accordingly, a second portion of the first screen 104 and a second portion of the second screen 108 remain available for the presentation of information other than the virtual keyboard 704 to a user.

[0101] Fig. 7B depicts a device 100 in a second mode or orientation in which first 104 and second 108 screens are in a dual landscape mode. In this example, the first screen 104 is above and adjacent to the second screen 108. In addition, a virtual keyboard 704 is presented by the second screen 108. As shown, the virtual keyboard 704 can occupy all of the area of the second screen 108. In addition, the area occupied by the virtual keyboard 704 can be the same or about the same (*e.g.*, within ±10%) while the screens 104, 108 are in either portrait or landscape modes. For example, one or two dimensions of the virtual keyboard 704 can be the same in the first and second orientations of the device 100. In accordance with still other embodiments, the virtual keyboard 704 can be presented by the first screen 104 while in a landscape mode. For example, where the device 100 is held or positioned such that the second screen 108 is above the first screen 104, the virtual keyboard 704 can be presented by the first screen 104. In accordance with still further embodiments of the present invention, a virtual keyboard 704 can be presented using portions of the first 104 and second 108 screens, even when those screens are in a landscape mode.

[0102] Fig. 8A presents details of an exemplary virtual keyboard 704, with a first portion 704a presented using a first portion of a first screen 104 and with a second portion 704b presented using a first portion of a second screen 108 of a device 100 in accordance with embodiments of the present invention. As shown in this example, and without limitation, the virtual keyboard 704 can comprise a QWERTY keyboard. Because the virtual keyboard 704 is presented using

example, the virtual keyboard 704 can be presented by the screen 104 or 108 that is below the other screen 104 or 108. After presenting the virtual keyboard 704 in the mode determined to be correct, a determination can be made as to whether the orientation of the device has changed (step 920). This can include whether an orientation of the device 100 has changed such that the device 100 has been rotated within a plane corresponding to or substantially containing the screens 104, 108 by 90°. If the orientation has changed, the process can return to step 908. Accordingly, even if the orientation of the device 100 has changed, the virtual keyboard 704 continues to be displayed. In addition, in accordance with embodiments of the present invention, the dimensions of the virtual keyboard 704 remain the same or about the same whether the device 100 is in dual portrait mode or is in dual landscape mode. If the orientation has not changed, a determination can next be made as to whether the content of the displayed output has changed (step 924). A change to the output can include the display of characters entered by a user through the virtual keyboard 704. In accordance with other embodiments, a change to the output can include a change to the information displayed through operation of application programming and/or the receipt of information at the device 100. A change in output can also include an acknowledgement of commands received from a user or from a connected device, or that are generated through operation of application programming, and the effect or processing of the command. After determining that the output provided by the displays 104, 108 has changed, the process can return to step 904, to determine whether, in view of the change or other information, the virtual keyboard 704 should continue to be displayed.

[0106] At step 928, a determination can be made as to whether operation of the device should continue, or whether the device has powered down. If operation is to continue, the process can return to step 904. If the device has been powered down, the process can end. Although steps in the process have been shown and described in a particular order, it should be appreciated that other orderings are possible and incorporated herein. For example, the process can be discontinued at any point in the process if the device 100 is powered off.

[0107] In accordance with embodiments of the present invention, the method disclosed herein can be performed by the execution of application programming or other software components stored in memory 208, 508 by a processor 204, 504. For instance, the operating system 516, a component or components of the framework 520, and/or a component or components of the multi-display management module 524 can include or provide functionality to receive and act on

two adjacent screens 104, 108, the line of separation 708 passes between a first portion 704a of the virtual keyboard 704 presented using the first screen 104, and a second portion 704b of the virtual keyboard 704 presented by the second screen 108. Moreover, because of the line of separation 708, the spacing of the keys 816 of the virtual keyboard 704 can be arranged such that the line of separation 708 falls between individual character keys 816. A space bar 820 can span across both the first 104 and second 108 screens.

[0103] Fig. 8B depicts a virtual keyboard 704 displayed by one of the screens 104 or 108 when the screens 104, 108 are in a landscape mode. In particular, in this example, the virtual keyboard 704 is displayed by the second screen 108, while the area of the first screen 104 remains available to present information other than the virtual keyboard 704 to a user. Moreover, the virtual keyboard occupies all of the second screen 108. As shown, in a dual landscape mode the virtual keyboard 704 can be presented in the screen 104, 108 that is below the other screen 104, 108.

[0104] Comparing the virtual keyboard 704 presented with portions of the screens 104, 108 in a portrait mode, such that the virtual keyboard 704 spans the two screens 104, 108, to the mode of operation in which the virtual keyboard 704 is presented by a single screen 104 or 108 in a landscape mode highlights changes that can be made to accommodate the line of separation 708. In particular, in the embodiment illustrated in Fig. 8A, the spacing of the keys 816 has been altered so that the smaller keys (*e.g.*, the character keys) lie on one side or the other of the line of separation 708. In contrast, in the embodiment illustrated in Fig. 8B, the keys 816 of the virtual keyboard 704 have a more conventional, interleaved type spacing.

[0105] Fig. 9 depicts aspects of a method for presenting a virtual keyboard 704 in accordance with embodiments of the present invention. Initially, at step 904, a determination is made as to whether a virtual keyboard is to be displayed. Until such a determination is made, the process may idle at step 904. After a determination to display a virtual keyboard 704 has been made, a determination can next be made as to the orientation of the device 100 (step 908). If the device 100 is in a portrait mode, the virtual keyboard 704 is presented using portions of both the first 104 and second 108 screens of the multiple screen device 100 (step 912). Accordingly, with the screens 104, 108 in the portrait mode, the virtual keyboard 704 spans portions of both screens 104, 108. If the device 100 screens 104, 108 are determined to be in a landscape orientation, the virtual keyboard 704 can be presented using one of the screens 104 or 108 (step 916). For

input received from a user in connection with the display and operation of a virtual keyboard 704. Moreover, such input can include input comprising user interface interactions by a user that are managed by the input management module 536, or other components. In addition, although certain embodiments have been described in connection with operation on a device having first 104 and second 108 screens, the invention is not limited to operation on such a device. For example, a device 100 can have more than two screens, and/or the screen can comprise virtual screens or windows.

[0108] The exemplary systems and methods of this disclosure have been described in relation to the presentation of a virtual keyboard 704 that in a dual portrait mode occupies portions of first and second screens 104 and 108, and that upon rotation to a dual landscape mode occupies all or substantially all of the bottom screen 104 or 108 of a device 100 with multiple screens 104, 108. However, to avoid unnecessarily obscuring the present disclosure, the preceding description omits a number of known structures and devices. This omission is not to be construed as a limitation of the scopes of the claims. Specific details are set forth to provide an understanding of the present disclosure. It should however be appreciated that the present disclosure may be practiced in a variety of ways beyond the specific detail set forth herein.

[0109] Furthermore, while the exemplary aspects, embodiments, and/or configurations illustrated herein show the various components of the system collocated, certain components of the system can be located remotely, at distant portions of a distributed network, such as a LAN and/or the Internet, or within a dedicated system. Thus, it should be appreciated, that the components of the system can be combined in to one or more devices, such as a computer, laptop, netbook, tablet computer, smart phone, mobile device, etc., or collocated on a particular node of a distributed network, such as an analog and/or digital telecommunications network, a packet-switch network, or a circuit-switched network. It will be appreciated from the preceding description, and for reasons of computational efficiency, that the components of the system can be arranged at any location within a distributed network of components without affecting the operation of the system. For example, the various components can be located in a switch such as a PBX and media server, gateway, in one or more communications devices, at one or more users' premises, or some combination thereof. Similarly, one or more functional portions of the system could be distributed between a telecommunications device(s) and an associated computing device.

[0110] Furthermore, it should be appreciated that the various links connecting the elements can be wired or wireless links, or any combination thereof, or any other known or later developed element(s) that is capable of supplying and/or communicating data to and from the connected elements. These wired or wireless links can also be secure links and may be capable of communicating encrypted information. Transmission media used as links, for example, can be any suitable carrier for electrical signals, including coaxial cables, copper wire and fiber optics, and may take the form of acoustic or light waves, such as those generated during radio-wave and infra-red data communications.

[0111] Also, while the flowcharts have been discussed and illustrated in relation to a particular sequence of events, it should be appreciated that changes, additions, and omissions to this sequence can occur without materially affecting the operation of the disclosed embodiments, configuration, and aspects.

[0112] A number of variations and modifications of the disclosure can be used. It would be possible to provide for some features of the disclosure without providing others.

[0113] For example in one alternative embodiment, the virtual keyboard 704 can be displayed within a portion of one screen 104 or 108 when the screens 104, 108 are in a landscape orientation.

[0114] In another alternative embodiment, in a landscape orientation, a first portion of the virtual keyboard 704 can be displayed by the first screen 104, while a second portion of the virtual keyboard 704 can be displayed by the second screen 108.

[0115] In yet another embodiment, the systems and methods of this disclosure can be implemented in conjunction with a special purpose computer, a programmed microprocessor or microcontroller and peripheral integrated circuit element(s), an ASIC or other integrated circuit, a digital signal processor, a hard-wired electronic or logic circuit such as discrete element circuit, a programmable logic device or gate array such as PLD, PLA, FPGA, PAL, special purpose computer, any comparable means, or the like. In general, any device(s) or means capable of implementing the methodology illustrated herein can be used to implement the various aspects of this disclosure. Exemplary hardware that can be used for the disclosed embodiments, configurations and aspects includes computers, handheld devices, telephones (e.g., cellular, Internet enabled, digital, analog, hybrids, and others), and other hardware known in the art. Some of these devices include processors (e.g., a single or multiple microprocessors), memory,

nonvolatile storage, input devices, and output devices. Furthermore, alternative software implementations including, but not limited to, distributed processing or component/object distributed processing, parallel processing, or virtual machine processing can also be constructed to implement the methods described herein.

[0116] In yet another embodiment, the disclosed methods may be readily implemented in conjunction with software using object or object-oriented software development environments that provide portable source code that can be used on a variety of computer or workstation platforms. Alternatively, the disclosed system may be implemented partially or fully in hardware using standard logic circuits or VLSI design. Whether software or hardware is used to implement the systems in accordance with this disclosure is dependent on the speed and/or efficiency requirements of the system, the particular function, and the particular software or hardware systems or microprocessor or microcomputer systems being utilized.

[0117] In yet another embodiment, the disclosed methods may be partially implemented in software that can be stored on a storage medium, executed on programmed general-purpose computer with the cooperation of a controller and memory, a special purpose computer, a microprocessor, or the like. In these instances, the systems and methods of this disclosure can be implemented as program embedded on personal computer such as an applet, JAVA® or CGI script, as a resource residing on a server or computer workstation, as a routine embedded in a dedicated measurement system, system component, or the like. The system can also be implemented by physically incorporating the system and/or method into a software and/or hardware system.

[0118] Although the present disclosure describes components and functions implemented in the aspects, embodiments, and/or configurations with reference to particular standards and protocols, the aspects, embodiments, and/or configurations are not limited to such standards and protocols. Other similar standards and protocols not mentioned herein are in existence and are considered to be included in the present disclosure. Moreover, the standards and protocols mentioned herein and other similar standards and protocols not mentioned herein are periodically superseded by faster or more effective equivalents having essentially the same functions. Such replacement standards and protocols having the same functions are considered equivalents included in the present disclosure.

[0119] The present disclosure, in various aspects, embodiments, and/or configurations, includes components, methods, processes, systems and/or apparatus substantially as depicted and described herein, including various aspects, embodiments, configurations embodiments, subcombinations, and/or subsets thereof. Those of skill in the art will understand how to make and use the disclosed aspects, embodiments, and/or configurations afrer understanding the present disclosure. The present disclosure, in various aspects, embodiments, and/or configurations, includes providing devices and processes in the absence of items not depicted and/or described herein or in various aspects, embodiments, and/or configurations hereof, including in the absence of such items as may have been used in previous devices or processes, e.g., for improving performance, achieving ease and\or reducing cost of implementation.

[0120] The foregoing discussion has been presented for purposes of illustration and description. The foregoing is not intended to limit the disclosure to the form or forms disclosed herein. In the foregoing Detailed Description for example, various features of the disclosure are grouped together in one or more aspects, embodiments, and/or configurations for the purpose of streamlining the disclosure. The features of the aspects, embodiments, and/or configurations of the disclosure may be combined in alternate aspects, embodiments, and/or configurations other than those discussed above. This method of disclosure is not to be interpreted as reflecting an intention that the claims require more features than are expressly recited in each claim. Rather, as the following claims reflect, inventive aspects lie in less than all features of a single foregoing disclosed aspect, embodiment, and/or configuration. Thus, the following claims are hereby incorporated into this Detailed Description, with each claim standing on its own as a separate preferred embodiment of the disclosure.

[0121] Moreover, though the description has included description of one or more aspects, embodiments, and/or configurations and certain variations and modifications, other variations, combinations, and modifications are within the scope of the disclosure, e.g., as may be within the skill and knowledge of those in the art, after understanding the present disclosure. It is intended to obtain rights which include alternative aspects, embodiments, and/or configurations to the extent permitted, including alternate, interchangeable and/or equivalent structures, functions, ranges or steps to those claimed, whether or not such alternate, interchangeable and/or equivalent structures, functions, ranges or steps are disclosed herein, and without intending to publicly dedicate any patentable subject matter.

What is claimed is:

1. A method, comprising:

providing a device having first and second screens;

positioning the device in a first device orientation;

in a first operating mode corresponding to the first device orientation, first presenting a virtual keyboard, wherein a first portion of the virtual keyboard is presented using a first portion of the first screen, and wherein a second portion of the virtual keyboard is presented using a first portion of the second screen;

receiving input from a user, the input including rotating the device to a second device orientation;

in a second operating mode corresponding to the second device orientation, second presenting a virtual keyboard, wherein all of the virtual keyboard is presented using one of the first screen and the second screen.

2. The method of claim 1, wherein in the first device orientation the first and second screens are in a portrait mode, and wherein in the second device orientation the first and second touch screens are in a landscape mode.

3. The method of claim 2, wherein the portrait mode is a dual portrait mode, and wherein the landscape mode is a dual landscape mode.

4. The method of claim 3, wherein in the first operating mode the method further comprises:

presenting using a second portion of the first touch screen first information.

5. The method of claim 4, further comprising:

presenting using a second portion of the second touch screen second information.

6. The method of claim 5, wherein in the second operating mode the method further comprises:

presenting using the second touch screen the first information and the second information.

7. The method of claim 1, wherein in the first operating mode the presentation of the virtual keyboard is modified as compared to the presentation of the virtual keyboard in the second operating mode, and wherein the modification includes aligning keys included in the virtual keyboard to accommodate a gap between a first group of the keys included in the virtual keyboard that are presented by the first touch screen and a second group of keys included in the virtual keyboard that are presented by the second touch screen.

8. The method of claim 1, wherein the virtual keyboard has at least one dimension that is the same in the first operating mode as in the second operating mode.

9. The method of claim 1, wherein the virtual keyboard has first and second dimensions that are the same in the first operating mode as in second operating mode.

10. The method of claim 1, wherein the virtual keyboard is a virtual alphanumeric keyboard.

11. A device, comprising:

a first screen, the first screen including a touch sensitive display area;

a second screen, the second screen including a touch sensitive display area;

memory;

a processor;

application programming stored in the memory and executed by the processor, wherein the application programming is operable to:

in a first operating mode, display a virtual keyboard using a portion of the touch sensitive display area of the first screen and using a portion of the touch sensitive display area of the second screen, wherein in the first operating mode the device is in a first device orientation;

detect a rotation of the device from the first device orientation to a second device orientation;

in a second operating mode, display the virtual keyboard using at least a portion of a first one of the touch sensitive display area of the first screen and the second screen, and without using any portion of a second one of the touch sensitive display area of the first screen and the second screen, wherein in the second operating mode the device is in the second device orientation.

12. The device of claim 11, wherein in the second operating mode all of the first one of the first screen and second screen is used to display the virtual keyboard.

13. The device of claim 11, further comprising:

an accelerometer, wherein the accelerometer is operative to provide an indication of an orientation of the first and second touch screens to the application programming, and wherein the first operating mode is entered while first screen is in a portrait orientation and while the second screen is in a portrait orientation.

14. The device of claim 13, wherein the virtual keyboard is in a first configuration in the first operating mode, and wherein the virtual keyboard is in a second configuration in the second operating mode.

15. The device of claim 14, wherein the virtual keyboard presents a QWERTY keyboard to a user, and wherein in the first operating mode a line between the first screen and the second screen extends between the T and the Y key, the G and the H key, and the V and the B key.

16. A computer readable medium having stored thereon computer-executable instructions, the computer executable instructions causing a processor to execute a method for providing a virtual keyboard, the computer-executable instructions comprising:

instructions to determine an orientation of a device having at least a first touch screen and a second touch screen;

instructions to display all of a virtual keyboard using a portion of a first touch screen and a portion of a second touch screen while the device is determined to be in a first orientation;

instructions to display all of the virtual keyboard using a first one of a first touch screen and a second touch screen while the device is determined to be in a second orientation.

17. The computer readable medium of claim 16, further comprising:

instructions to determine a change in the orientation of the device.

18. The computer readable medium of claim 17, further comprising:

instructions to display a portion of first information using a second portion of the first touch screen and to display a portion of the first information using a second portion of the second touch screen while the first and second touch screens are determined to be in the portrait mode.

19. The computer readable medium of claim 18, further comprising:

instructions to display a first portion of a virtual keyboard using a first portion of a first touch screen and to display a second portion of the virtual keyboard using a second portion of a second touch screen.

20. The computer readable medium of claim 19, wherein at least some keys of the virtual keyboard with the first and second touch screens determined to be in the portrait orientation are shifted to accommodate a border between the first and second touch screens as compared to the keys of the virtual keyboard with the first and second touch screens determined to be in the landscape orientation.

ABSTRACT

Methods and devices for presenting a virtual keyboard are provided. More particularly, in connection with a multiple screen device, a virtual keyboard can be presented in a first mode using portions of both of the screens. In a second mode, the virtual keyboard can be presented using all of one of the screens. Movement between the different modes can be effected by rotating the device between a dual portrait orientation, corresponding to the first mode, and a dual landscape orientation, corresponding to the second mode. More particularly, with the screens of the device in a landscape orientation, one screen can be devoted to present the virtual keyboard while the other screen remains available to present other information.

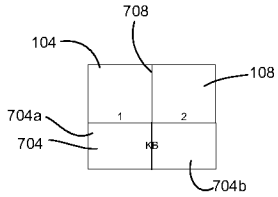


Fig. 7A

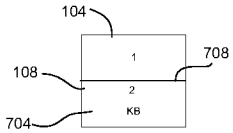


Fig. 7B

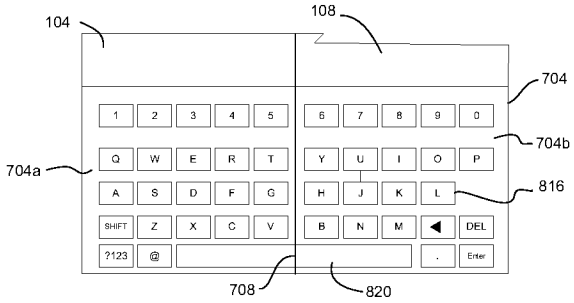


Fig. 8A

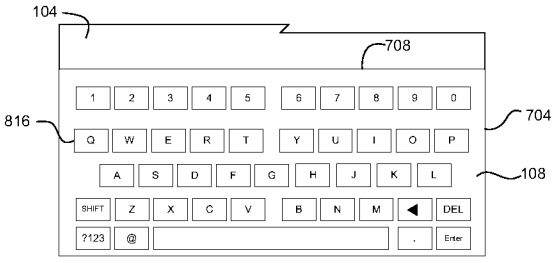


Fig. 8B

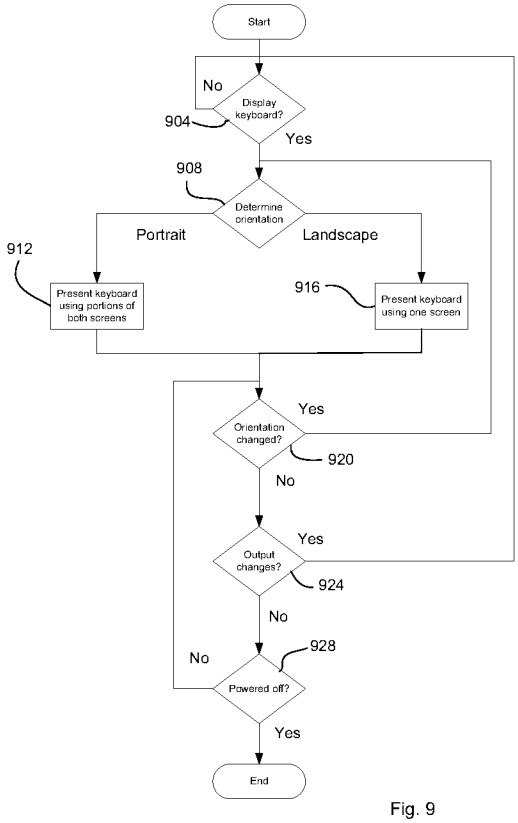


Fig. 9

GRAVITY DROP RULES AND KEYBOARD DISPLAY ON A MULTIPLE SCREEN DEVICE

[0100] Fig. 7A depicts a device 100 with first 104 and second 108 screens. In this example, the device 100 is depicted in a first mode or orientation in which the screens 104, 108 are in a dual portrait mode. A virtual keyboard 704 is presented by the screens 104, 108. In particular, a first portion 704a of the virtual keyboard 704 is presented by a first portion 104a of the first screen 104, while a second portion 704b of the virtual keyboard 704 is presented by a first portion 108a of the second screen 108. More particularly, the virtual keyboard 704 spans a line of separation 708 between the first and second screens. Moreover, the virtual keyboard 704 does not occupy all of the area of either the first screen 104 or the second screen 108. Accordingly, a second portion 104b of the first screen 104 and a second portion 108b of the second screen 108 remain available for the presentation of information other than the virtual keyboard 704 to a user. In this example, the second portion 104b of the first screen 104 presents at least a portion of a first application or set of information in or as part of a first window 712, while the second portion 108b of the second screen 108 displays at least a portion of a second application or set of information in or as part of a second window 716. In addition, an input field 720 is presented by the first screen 104. The association of the input field 720 with the first application 712 indicates that the focus of the user is on the first application 712, and not the second application 716. Although the provision of an input field 720 is used in various examples included herein to designate the screen 104 or 108 displaying information 712 or 716 having the focus of the user, it should be appreciated that the inclusion of an input field 720 is not a necessary indication of the focus of the user. For instance, in accordance with alternate embodiments, the focus of a user can be indicated through a selection by a set of information 712 or 716 by a user, or through the application of gravity drop rules, without necessarily including or providing an input field 720.

[0101] Fig. 7B depicts the device 100 in a second mode or orientation in which first 104 and second 108 screens are in a dual landscape mode. In this example, the first screen 104 is above and adjacent to the second screen 108. In addition, the orientation depicted in Fig. 7B can be entered or achieved, starting with the orientation depicted in Fig. 7A, afrer rotating the device 100 in a clockwise direction. As used herein, the clockwise direction is determined from the perspective of a user viewing the output of the screens 104, 108 of the device 100. A rotation of the device 100 in a clockwise direction by 90° starting from the configuration illustrated in Fig. 7A has the effect of placing the first screen 104 above the second screen 108. This is taken as an indication that the first application or set of information 712 is of primary interest to the user. Moreover, because the first application 712 has the focus of the user, as indicated in the example figure by the inclusion of an input field 720, the virtual keyboard 704 continues to be presented to the user. In addition, the input field 720 continues to be presented to the user. In the second orientation illustrated in Fig. 7B, the virtual keyboard 704 fills all of a portion of the second screen 108. In addition, since the user has, through the rotation of the device 100 to place the first application or information 712 in the upper or primary screen while the focus of the user is on that application 712, gravity drop rules can be applied to specify that, in response to input from a user dismissing the keyboard 704, the first application 712 is expanded to fill the second screen 108 where the first application 712 is a dual or multiple screen application.

[0102] Fig. 7C depicts the device 100 in yet another mode or orientation. In particular, the device 100 has been rotated, starting from the orientation and configuration depicted in Fig. 7A, in a counterclockwise direction by 90°, such that the second screen 108 is positioned above the first screen 104 in a dual landscape mode. Accordingly, the device 100, as configured in Fig. 7A, has been rotated in a direction towards the application or information 712 having the focus of the user. In accordance with embodiments of the present invention, this rotation of the device is taken as an indication that the user is interested in the second application 716. Therefore, since the virtual keyboard 704 is associated with the provision of input, for example as indicated by the input field 720, for the first application 712, rotation in the counterclockwise direction in this example results in the dismissal or discontinuance of the presentation of the virtual keyboard 704. Moreover, the second application 716 is expanded across both screens 104, 108, where the second application 716 is a dual or multiple screen application.

[0103] Fig. 8A depicts the device 100 with first 104 and second 108 screens displaying first 712 and second 716 sets of information or applications respectively. In this example, the focus of the user is on the second application 716, displayed by the second screen 108, as indicated by the inclusion of a user input window 720. In addition, a virtual keyboard 704 is displayed, such that input to the second application 716 can be entered using the virtual keyboard 704.

[0104] Fig. 8B depicts the device 100 in a second mode or orientation in which the first screen 104 is positioned above the second screen 108, with those screens 104, 108 in a dual landscape mode. Starting from the configuration depicted in Fig. 8A, the configuration illustrated in Fig.

8B is entered or obtained afrer input from the user that comprises a rotation of the device 100 in a clockwise direction. A clockwise rotation of the device 100 as configured in Fig. 8A moves the first application 712, which does not have the focus of the user, into the top or primary screen 104. This movement is taken as an indication that the user is no longer interested in providing input with respect to the second application 716. Accordingly, rotation of the device 100 in the clockwise direction by 90° from the configuration illustrated in Fig. 8A results in the dismissal or discontinuation of the presentation of the virtual keyboard 704. Moreover, the first application 712 is expanded to both the first 104 and second 108 screens, where the first application 712 is a dual or multiple screen application.

[0105] Fig. 8C depicts the output of the device 100 afrer that device 100, configured as illustrated in Fig. 8A, has been illustrated in a counterclockwise direction. In particular, rotation of the device in a counterclockwise direction by 90° places the second application 716, which has the focus of the user, in the uppermost or primary screen. In response, the virtual keyboard 704 continues to be presented, while the second application 716 is displayed by the second screen 108. In addition, the input field 720 continues to be presented. From the configuration illustrated in Fig. 8C, if the virtual keyboard 704 is dismissed, for example as a result of user input, additional portions of the second application 716, for example a second page, is revealed.

[0106] Although various examples provided herein have illustrated a starting configuration with a first screen 104 on the left and a second screen 108 on the right, other starting configurations are possible. For instance, the starting orientation can have the second screen 108 on the lefr and the first screen 104 on the right when the device 100 is in a dual portrait format or orientation.

[0107] Fig. 9 depicts aspects of a method for presenting a virtual keyboard 704 in accordance with embodiments of the present invention. Initially, at step 904, a virtual keyboard 704 is presented. More particularly, with the device 100 in a dual portrait mode, portions of the first 104 and second 108 screens are used to display the virtual keyboard 704. At step 912, input is received changing the orientation of the device 100. In particular, the received input comprises rotation of the device by 90° in either the clockwise or counterclockwise direction. At step 916, a determination is made as to whether the rotation of the device 100 is in the direction of the application or information having the current focus of the user (step 916). The focus of the user can be determined from the application or information that has the input focus of the user. In

particular, the input focus of the user is in place with respect to an application 712 or 716 when an input mode is active with respect to that application 712 or 716. In accordance with other embodiments of the present invention, the focus of the user can be determined from the selection of a window or portions of an output associated with a particular application or set of information 712 or 716.

[0108] If the rotation of the device 100 is in the direction of the application or information 712 or 716 having the focus of the user, the virtual keyboard 704 is dismissed, and the information or application 712 or 716 that did not have the focus of the user when the device 100 was rotated is expanded. In particular, the rotation of the device in the direction of the application or information 712 or 716 that have the focus of the user has the effect of moving the other information 712 or 716 into the upper screen 104 or 108 of the device 100, which is taken as an indication that the user is primarily interested in that other information 712 or 716. Moreover, because the focus of the user is no longer on the application or information 712 or 716 that previously had input focus, the virtual keyboard 704 is dismissed.

[0109] If the rotation of the device 100 is not in the direction of the application or information 712 or 716 with the focus of the user, the virtual keyboard continues to be displayed (step 924). At step 928, a determination can be made as to whether the virtual keyboard 704 should be dismissed, for example as a result of input received from the user. If the virtual keyboard 704 should be dismissed, the information 712 or 716 presented by the upper screen 104 or 108 is, upon the dismissal of the virtual keyboard 704, expanded into the lower screen 104 or 108 (step 932).

[0110] In accordance with embodiments of the present invention, the method disclosed herein can be performed by the execution of application programming or other software components stored in memory 208, 508 by a processor 204, 504. For instance, the operating system 516, a component or components of the framework 520, and/or a component or components of the multi-display management module 524 can include or provide functionality to receive and act on input received from a user in connection with the display and operation of a virtual keyboard 704. Moreover, such input can include input comprising user interface interactions by a user that are managed by the input management module 536, or other components. In addition, although certain embodiments have been described in connection with operation on a device having first 104 and second 108 screens, the invention is not limited to operation on such a device. For

example, a device 100 can have more than two screens, and/or the screen can comprise virtual screens or windows.

[0111] The exemplary systems and methods of this disclosure have been described in relation to the presentation of a virtual keyboard 704 that in a dual portrait mode occupies portions of first and second screens 104 and 108, and that upon rotation to a dual landscape mode either occupies all or substantially all of the bottom screen 104 or 108 of a device 100 with multiple screens 104, 108, or is dismissed. However, to avoid unnecessarily obscuring the present disclosure, the preceding description omits a number of known structures and devices. This omission is not to be construed as a limitation of the scopes of the claims. Specific details are set forth to provide an understanding of the present disclosure. It should however be appreciated that the present disclosure may be practiced in a variety of ways beyond the specific detail set forth herein.

[0112] Furthermore, while the exemplary aspects, embodiments, and/or configurations illustrated herein show the various components of the system collocated, certain components of the system can be located remotely, at distant portions of a distributed network, such as a LAN and/or the Internet, or within a dedicated system. Thus, it should be appreciated, that the components of the system can be combined in to one or more devices, such as a computer, laptop, netbook, tablet computer, smart phone, mobile device, etc., or collocated on a particular node of a distributed network, such as an analog and/or digital telecommunications network, a packet-switch network, or a circuit-switched network. It will be appreciated from the preceding description, and for reasons of computational efficiency, that the components of the system can be arranged at any location within a distributed network of components without affecting the operation of the system. For example, the various components can be located in a switch such as a PBX and media server, gateway, in one or more communications devices, at one or more users' premises, or some combination thereof. Similarly, one or more functional portions of the system could be distributed between a telecommunications device(s) and an associated computing device.

[0113] Furthermore, it should be appreciated that the various links connecting the elements can be wired or wireless links, or any combination thereof, or any other known or later developed element(s) that is capable of supplying and/or communicating data to and from the connected elements. These wired or wireless links can also be secure links and may be capable of

communicating encrypted information. Transmission media used as links, for example, can be any suitable carrier for electrical signals, including coaxial cables, copper wire and fiber optics, and may take the form of acoustic or light waves, such as those generated during radio-wave and infra-red data communications.

[0114] Also, while the flowcharts have been discussed and illustrated in relation to a particular sequence of events, it should be appreciated that changes, additions, and omissions to this sequence can occur without materially affecting the operation of the disclosed embodiments, configuration, and aspects.

[0115] A number of variations and modifications of the disclosure can be used. It would be possible to provide for some features of the disclosure without providing others.

[0116] For example in one alternative embodiment, the virtual keyboard 704 can be displayed within a portion of one screen 104 or 108 when the screens 104, 108 are in a landscape orientation.

[0117] In another alternative embodiment, in a landscape orientation, a first portion of the virtual keyboard 704 can be displayed by the first screen 104, while a second portion of the virtual keyboard 704 can be displayed by the second screen 108.

[0118] In yet another embodiment, the systems and methods of this disclosure can be implemented in conjunction with a special purpose computer, a programmed microprocessor or microcontroller and peripheral integrated circuit element(s), an ASIC or other integrated circuit, a digital signal processor, a hard-wired electronic or logic circuit such as discrete element circuit, a programmable logic device or gate array such as PLD, PLA, FPGA, PAL, special purpose computer, any comparable means, or the like. In general, any device(s) or means capable of implementing the methodology illustrated herein can be used to implement the various aspects of this disclosure. Exemplary hardware that can be used for the disclosed embodiments, configurations and aspects includes computers, handheld devices, telephones (e.g., cellular, Internet enabled, digital, analog, hybrids, and others), and other hardware known in the art. Some of these devices include processors (e.g., a single or multiple microprocessors), memory, nonvolatile storage, input devices, and output devices. Furthermore, alternative software implementations including, but not limited to, distributed processing or component/object distributed processing, parallel processing, or virtual machine processing can also be constructed to implement the methods described herein.

[0119] In yet another embodiment, the disclosed methods may be readily implemented in conjunction with software using object or object-oriented software development environments that provide portable source code that can be used on a variety of computer or workstation platforms. Alternatively, the disclosed system may be implemented partially or fully in hardware using standard logic circuits or VLSI design. Whether software or hardware is used to implement the systems in accordance with this disclosure is dependent on the speed and/or efficiency requirements of the system, the particular function, and the particular software or hardware systems or microprocessor or microcomputer systems being utilized.

[0120] In yet another embodiment, the disclosed methods may be partially implemented in software that can be stored on a storage medium, executed on programmed general-purpose computer with the cooperation of a controller and memory, a special purpose computer, a microprocessor, or the like. In these instances, the systems and methods of this disclosure can be implemented as program embedded on personal computer such as an applet, JAVA® or CGI script, as a resource residing on a server or computer workstation, as a routine embedded in a dedicated measurement system, system component, or the like. The system can also be implemented by physically incorporating the system and/or method into a software and/or hardware system.

[0121] Although the present disclosure describes components and functions implemented in the aspects, embodiments, and/or configurations with reference to particular standards and protocols, the aspects, embodiments, and/or configurations are not limited to such standards and protocols. Other similar standards and protocols not mentioned herein are in existence and are considered to be included in the present disclosure. Moreover, the standards and protocols mentioned herein and other similar standards and protocols not mentioned herein are periodically superseded by faster or more effective equivalents having essentially the same functions. Such replacement standards and protocols having the same functions are considered equivalents included in the present disclosure.

[0122] The present disclosure, in various aspects, embodiments, and/or configurations, includes components, methods, processes, systems and/or apparatus substantially as depicted and described herein, including various aspects, embodiments, configurations embodiments, subcombinations, and/or subsets thereof. Those of skill in the art will understand how to make and use the disclosed aspects, embodiments, and/or configurations afrer understanding the

present disclosure. The present disclosure, in various aspects, embodiments, and/or configurations, includes providing devices and processes in the absence of items not depicted and/or described herein or in various aspects, embodiments, and/or configurations hereof, including in the absence of such items as may have been used in previous devices or processes, e.g., for improving performance, achieving ease and\or reducing cost of implementation.

[0123] The foregoing discussion has been presented for purposes of illustration and description. The foregoing is not intended to limit the disclosure to the form or forms disclosed herein. In the foregoing Detailed Description for example, various features of the disclosure are grouped together in one or more aspects, embodiments, and/or configurations for the purpose of streamlining the disclosure. The features of the aspects, embodiments, and/or configurations of the disclosure may be combined in alternate aspects, embodiments, and/or configurations other than those discussed above. This method of disclosure is not to be interpreted as reflecting an intention that the claims require more features than are expressly recited in each claim. Rather, as the following claims reflect, inventive aspects lie in less than all features of a single foregoing disclosed aspect, embodiment, and/or configuration. Thus, the following claims are hereby incorporated into this Detailed Description, with each claim standing on its own as a separate preferred embodiment of the disclosure.

[0124] Moreover, though the description has included description of one or more aspects, embodiments, and/or configurations and certain variations and modifications, other variations, combinations, and modifications are within the scope of the disclosure, e.g., as may be within the skill and knowledge of those in the art, afrer understanding the present disclosure. It is intended to obtain rights which include alternative aspects, embodiments, and/or configurations to the extent permitted, including alternate, interchangeable and/or equivalent structures, functions, ranges or steps to those claimed, whether or not such alternate, interchangeable and/or equivalent structures, functions, ranges or steps are disclosed herein, and without intending to publicly dedicate any patentable subject matter.

What is claimed is:

1. A method, comprising:

providing a device having first and second screens;

positioning the device in a first device orientation;

presenting a virtual keyboard, wherein a first portion of the virtual keyboard is presented using a first portion of the first screen, wherein a second portion of the virtual keyboard is presented using a first portion of the second screen, wherein first information is presented using a second portion of the first screen, and wherein second information is presented using a second portion of the second screen;

receiving input from a user, the input including rotating the device to a second device orientation;

determining that a focus of the user is on the first information presented by the first screen;

determining a direction of the rotation of the device;

the method further including at least one of:

in response to determining that the direction of rotation of the device is towards the first screen:

discontinuing presenting the virtual keyboard;

discontinuing presenting the first information;

expanding the presentation of the second information, wherein after expanding the presentation of the second information the first and second screens at least partially display at least one of the second information or information related to the second information;

in response to determining that the direction of rotation of the device is away from the first screen:

continuing to present the virtual keyboard;

discontinuing presenting the second information;

continuing to present the first information.

2. The method of claim 1, wherein in the first orientation the first and second screens are in a portrait mode, and wherein in the second operating mode the first and second touch screens are in a landscape mode.

3. The method of claim 2, wherein the direction of rotation is towards the first screen.

4. The method of claim 2, wherein the direction of rotation is towards the second screen.

5. The method of claim 4, further comprising:

dismissing the keyboard;

afrer dismissing the keyboard, presenting using the second screen at least one of an expansion of the second information or information related to the second information.

6. The method of claim 5, further comprising:

applying gravity drop rules to determine the information presented using the second screen after dismissing the keyboard.

7. The method of claim 4, wherein in the first device orientation the first screen is to the lefr of the second screen.

8. The method of claim 4, wherein in the first device orientation the first screen is to the right of the second screen.

9. The method of claim 1, wherein in the first device orientation the first information is presented in a window occupying all or substantially all of the first screen and the second information is presented in a window occupying all or substantially all of the second screen.

10. The method of claim 1, wherein in the first device orientation the presentation of the virtual keyboard is modified as compared to the presentation of the virtual keyboard in the

second device orientation, and wherein the modification includes aligning keys included in the virtual keyboard to accommodate a gap between a first group of the keys included in the virtual keyboard that are presented by the first touch screen and a second group of keys included in the virtual keyboard that are presented by the second touch screen.

11. A device, comprising:

a first screen, the first screen including a touch sensitive display area;

a second screen, the second screen including a touch sensitive display area;

memory;

a processor;

application programming stored in the memory and executed by the processor, wherein the application programming is operable to:

in a first device orientation, display a virtual keyboard using a portion of the touch sensitive display area of the first screen and using a portion of the touch sensitive display area of the second screen;

identify one of the first and second screens as having a focus of a user;

detect a rotation of the device from the first device orientation to a second device orientation;

in response to detecting a rotation of the device from the first device orientation to the second device orientation:

determining whether the direction of rotation is towards the one of the first and second screens identified as having the focus of the user, and one of:

in response to determining that the direction of rotation is not towards the one of the first and second screens identified as having the focus of the user, continuing the display of the virtual keyboard;

in response to determining that the direction of rotation is towards the one of the first and second screens identified as having the focus of the user, discontinuing the display of the virtual keyboard.

12. The device of claim 11, wherein the direction of rotation is not towards the first one of the first and second screens identified as having the focus of the user.

13. The device of claim 12, further comprising:

in the second device orientation, displaying the virtual keyboard using the second screen;

receiving a user input to discontinue the display of the virtual keyboard;

in response to the user input to discontinue the display of the virtual keyboard, displaying using the second screen information related to information displayed by the first screen.

14. The device of claim 11, wherein the direction of rotation is towards the first one of the first and second screens identified as having the focus of the user, wherein the application programming is further operable to display first information on the first and second screens.

15. The device of claim 11, further comprising:

an accelerometer, wherein the accelerometer is operative to provide an indication of an orientation of the first and second touch screens and a direction of rotation to the application programming.

16. A computer readable medium having stored thereon computer-executable instructions, the computer executable instructions causing a processor to execute a method for providing a virtual keyboard, the computer-executable instructions comprising:

instructions to display a first portion of a virtual keyboard using a first portion of a first screen of a multiple screen device, to display first information using a second portion of the first screen, to display a second portion of the virtual keyboard using a first portion of a second screen of the multiple screen device, and to display second information using a second portion of the second screen while the multiple screen device is in a first device orientation;

instructions to determine the focus of a user of the multiple screen device, wherein the focus is on one of the first information and the second information;

instructions to determine whether a direction of rotation of the multiple screen device from the first device orientation to the second device orientation is towards the screen containing the first information or the second information;

instructions, in response to determining that the direction of rotation of the multiple screen device is towards the screen containing the second information, and where the focus of the user is on the first information, to:

display the first information on a top one of the first and second screens;

display the virtual keyboard on a bottom one of the first and second screens.

17. The computer readable medium of claim 16, further comprising:

instructions, in response to determining that the direction of rotation of the multiple screen device is towards the screen containing the second information, and where the focus of the user is on the second information, to:

discontinue the display of the virtual keyboard;

display the first information and an extension of the first information on the first and second screens.

18. The computer readable medium of claim 16, further comprising:

instructions to discontinue the display of the virtual keyboard in response to input from the user;

instructions to display on the bottom one of the first and second screens an extension of the first information.

19. The computer readable medium of claim 17, further comprising:

instructions to determine a focus of the user from a user selection of one of the first and second information.

20. The computer readable medium of claim 17, wherein a first page of the first information is displayed by the first screen and a second page of the first information is displayed by the second screen.

ABSTRACT

Methods and devices for presenting or dismissing a virtual keyboard are provided. More particularly, in connection with a multiple screen device, a virtual keyboard can be presented in a first mode using portions of both of the screens. In a second mode, the virtual keyboard can be presented using all of one of the screens, or can be dismissed. Movement between the different modes can be effected by rotating the device between a dual portrait orientation, corresponding to the first mode, and a dual landscape orientation, corresponding to the second mode. More particularly, when the device is rotated in a direction away from information having the focus of the user, a display of the virtual keyboard can be continued. In contrast, when the device is rotated towards an application having the focus of the user, the display of the virtual keyboard can be discontinued, and instead additional portions of information that did not have the focus of the user is presented.

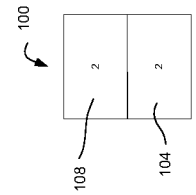


Fig. 7C

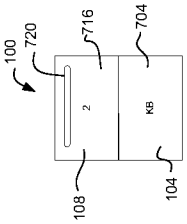


Fig. 8C

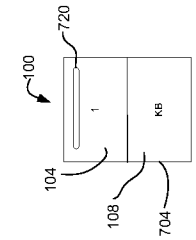


Fig. 7B

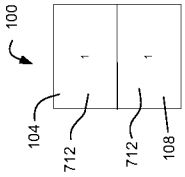


Fig. 8B

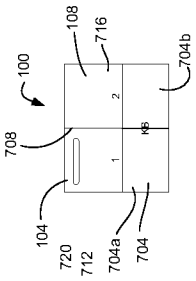


Fig. 7A

104

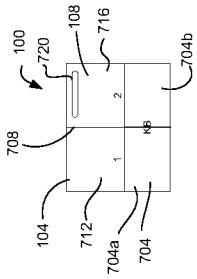


Fig. 8A

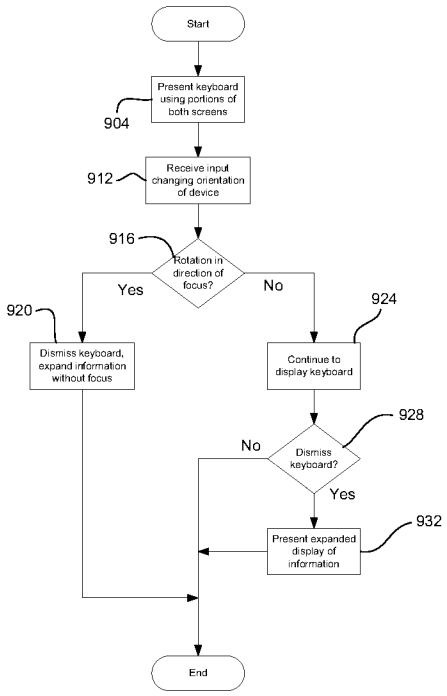


Fig. 9

KEYBOARD OPERATION ON APPLICATION LAUNCH

[0100] Fig. 7A depicts a device 100 with first 104 and second 108 screens. A virtual keyboard 704 is presented by the screens 104, 108. In particular, a first portion 704a of the virtual keyboard 704 is presented by a first portion 104a of the first screen 104, while a second portion 704b of the virtual keyboard 704 is presented by a first portion 108a of the second screen 108. More particularly, the virtual keyboard 704 spans a line of separation 708 between the first and second screens. Moreover, the virtual keyboard 704 does not occupy all of the area of either the first screen 104 or the second screen 108. Accordingly, a second portion 104b of the first screen 104 and a second portion 108b of the second screen 108 remain available for the presentation of information other than the virtual keyboard 704 to a user. More particularly, in this example, a first application or window 712 is displayed by a portion of the first screen 104, while a second application or window 714, here comprising a desktop, is displayed by a portion of the second screen 108. In addition, the first application or window 712 has keyboard focus, as indicated by the input field 720. As used herein, keyboard focus means that a virtual keyboard or other input means is operable with respect to the application having the keyboard focus, allowing a user to provide input to the application or that affects the application through the virtual keyboard 704 or other input.

[0101] Fig. 7B depicts the output of the device 100 illustrated in Fig. 7A, after the launch of an application 716, presented by the second screen 108. The original application 712 is presented by the first screen 104. In this example, the newly launched application 716, is not set to receive keyboard focus on launch. Accordingly, launch of the application 716 results in the dismissal of the keyboard 704. In addition, the input field 720 in the configuration illustrated in Fig. 7A can be discontinued.

[0102] Fig. 8A depicts a device 100 on which a virtual keyboard 704 is presented by first 104 and second 108 screens. A first application or window 712 is presented by the first screen 104, while a second window 714, in this example a second desktop, is displayed by the second screen 108. In Fig. 8B, the device 100 shown in Fig. 8A is illustrated, after the launch of an application set to receive keyboard focus on launch. Accordingly, as shown in Fig. 8B, the second screen 108 now displays a newly launched application or window 716, here Application 2. In addition, the virtual keyboard 704 continues to be displayed. Notably, whereas an input field 720 was presented in connection with the first application 712 in the initial state illustrated in Fig. 8A,

after the second application 716 is launched, the input field 720 is presented in connection with the second application 716. Accordingly, by providing instructions to present a virtual keyboard 704 on launch, the second application 716 is ready to receive input from the user, without requiring the user to take additional steps to call up the virtual keyboard 704.

[0103] Fig. 9A depicts a device 100 in another exemplary scenario. In this example, a first application 712 is presented on a first screen 104, and a second window 714, here a desktop view, is presented by the second screen 108. No virtual keyboard is presented as part of the initial state. In Fig. 9B, the device 100 illustrated in Fig. 9A is shown, after a second application 716 has been launched that is set to receive keyboard focus on launch. Accordingly, after the second application 716 is launched, the device 100 displays a virtual keyboard 704. Moreover, an input field 720 can be presented in connection with the second application 716. Accordingly, embodiments of the present invention provide for the presentation of a virtual keyboard 704 to enable a user to provide input upon the launch of an application 716, without requiring any additional user input.

[0104] Fig. 10 depicts aspects of a method for providing a virtual keyboard on application launch or initiation in accordance with embodiments of the present invention. Initially, at step 1004, input is received to launch an application 712, 716. As an example, such input can be received as user input, for example selecting an application to launch. As a further example, input to launch an application can be received as an instruction from another application, executed locally or remotely.

[0105] At step 1008, a determination is made as to whether the application being launched is set to receive keyboard focus on launch. If the application being launched is set to receive keyboard focus, a determination can next be made as to whether a virtual keyboard 704 is already being presented by the device 100 (step 1012). If a virtual keyboard is already being presented, that virtual keyboard 704 can continue to be presented by the device 100, in addition to the presentation of the application being launched (step 1016). If at step 1012 it is determined that the virtual keyboard 704 is not already being presented, a virtual keyboard 704 can be presented, together with the launched application (step 1020).

[0106] If at step 1008 it is determined that the application being launched is not set to receive keyboard focus on launch, a determination is made as to whether a virtual keyboard 704 is already being presented (step 1024). If the virtual keyboard 704 is already being presented, the

presentation of the virtual keyboard 704 is discontinued (step 1028). After discontinuing the presentation of the virtual keyboard, or after determining that a virtual keyboard is not already present, the launched application is presented (step 1032). Accordingly, in connection with the launch of an application that is not set to receive keyboard focus, no virtual keyboard 704 is displayed.

[0107] As can be appreciated by one of skill in the art after consideration of the present disclosure, embodiments of the present invention enable a virtual keyboard 704 to be presented or not, depending on settings associated with an application being launched. More particularly, where the application being launched is associated with an instruction place keyboard focus on the application on launch, a virtual keyboard 704 can either be generated and presented with the application being launched, or where a virtual keyboard 704 already exists, that existing virtual keyboard 704 can be maintained when the launched application is presented. Conversely, where an application being launched is not associated with an instruction to associate a keyboard focus with the application, the launch of the application does not result in a concurrent presentation of a virtual keyboard 704. In addition, where a virtual keyboard 704 is being presented at the time an instruction to launch an application that is not set to receive keyboard focus on launch is received, the presentation of that virtual keyboard 704 can be discontinued upon the launch of the application. In accordance with embodiments of the present invention, the instructions to set keyboard focus on an application (or not) on launch can be implemented in various ways. For example, such instructions can be provided as part of the executable code of the application, as a system setting, or as a script that is executed upon the launch of the application. Moreover, the determination as to whether or not an application being launched is set to receive a keyboard focus can be implemented in various ways, such as by operating system 516 and/or ftamework 520 components.

[0108] Although various examples and illustrations have shown a device 100 having first 104 and second 108 screens that include touch screen portions, embodiments of the present invention are not limited to such embodiments. For example, a device 100 incorporating a single touch screen divided into multiple windows can implement embodiments of the invention disclosed herein. In addition, although various examples and illustrations have shown a device with multiple screens 104 and 108 in a dual portrait orientation, embodiments of the present invention can be applied in connection with a dual landscape orientation. Moreover, a device 100 having a

single display screen can also implement embodiments of the present invention where multiple windows, for example in a dual portrait, dual landscape, or other orientation or arrangement capable of displaying one or more application windows together with a virtual keyboard are provided.

[0109] In accordance with embodiments of the present invention, the method disclosed herein can be performed by the execution of application programming or other software components stored in memory 208, 508 by a processor 204, 504. For instance, the operating system 516, a component or components of the framework 520, and/or a component or components of the multi-display management module 524 can include or provide functionality to receive and act on input received ftom a user in connection with the display and operation of a virtual keyboard 704. Moreover, such input can include input comprising user interface interactions by a user that are managed by the input management module 536, or other components. In addition, although certain embodiments have been described in connection with operation on a device having first 104 and second 108 screens, the invention is not limited to operation on such a device. For example, a device 100 can have more than two screens, and/or the screen can comprise virtual screens or windows.

[0110] The exemplary systems and methods of this disclosure have been described in relation to the selective presentation of a virtual keyboard 704 upon the receipt of instructions to launch an application. However, to avoid unnecessarily obscuring the present disclosure, the preceding description omits a number of known structures and devices. This omission is not to be construed as a limitation of the scopes of the claims. Specific details are set forth to provide an understanding of the present disclosure. It should however be appreciated that the present disclosure may be practiced in a variety of ways beyond the specific detail set forth herein.

[0111] Furthermore, while the exemplary aspects, embodiments, and/or configurations illustrated herein show the various components of the system collocated, certain components of the system can be located remotely, at distant portions of a distributed network, such as a LAN and/or the Internet, or within a dedicated system. Thus, it should be appreciated, that the components of the system can be combined in to one or more devices, such as a computer, laptop, netbook, tablet computer, smart phone, mobile device, etc., or collocated on a particular node of a distributed network, such as an analog and/or digital telecommunications network, a packet-switch network, or a circuit-switched network. It will be appreciated ftom the preceding

description, and for reasons of computational efficiency, that the components of the system can be arranged at any location within a distributed network of components without affecting the operation of the system. For example, the various components can be located in a switch such as a PBX and media server, gateway, in one or more communications devices, at one or more users' premises, or some combination thereof. Similarly, one or more functional portions of the system could be distributed between a telecommunications device(s) and an associated computing device.

[0112] Furthermore, it should be appreciated that the various links connecting the elements can be wired or wireless links, or any combination thereof, or any other known or later developed element(s) that is capable of supplying and/or communicating data to and from the connected elements. These wired or wireless links can also be secure links and may be capable of communicating encrypted information. Transmission media used as links, for example, can be any suitable carrier for electrical signals, including coaxial cables, copper wire and fiber optics, and may take the form of acoustic or light waves, such as those generated during radio-wave and infra-red data communications.

[0113] Also, while the flowcharts have been discussed and illustrated in relation to a particular sequence of events, it should be appreciated that changes, additions, and omissions to this sequence can occur without materially affecting the operation of the disclosed embodiments, configuration, and aspects.

[0114] A number of variations and modifications of the disclosure can be used. It would be possible to provide for some features of the disclosure without providing others.

[0115] For example in one alternative embodiment, the virtual keyboard 704 can be displayed within a portion of one screen 104 or 108 when the screens 104, 108 are in a landscape orientation.

[0116] In another alternative embodiment, in a landscape orientation, a first portion of the virtual keyboard 704 can be displayed by the first screen 104, while a second portion of the virtual keyboard 704 can be displayed by the second screen 108.

[0117] In yet another embodiment, the systems and methods of this disclosure can be implemented in conjunction with a special purpose computer, a programmed microprocessor or microcontroller and peripheral integrated circuit element(s), an ASIC or other integrated circuit, a digital signal processor, a hard-wired electronic or logic circuit such as discrete element circuit,

a programmable logic device or gate array such as PLD, PLA, FPGA, PAL, special purpose computer, any comparable means, or the like. In general, any device(s) or means capable of implementing the methodology illustrated herein can be used to implement the various aspects of this disclosure. Exemplary hardware that can be used for the disclosed embodiments, configurations and aspects includes computers, handheld devices, telephones (e.g., cellular, Internet enabled, digital, analog, hybrids, and others), and other hardware known in the art. Some of these devices include processors (e.g., a single or multiple microprocessors), memory, nonvolatile storage, input devices, and output devices. Furthermore, alternative software implementations including, but not limited to, distributed processing or component/object distributed processing, parallel processing, or virtual machine processing can also be constructed to implement the methods described herein.

[0118] In yet another embodiment, the disclosed methods may be readily implemented in conjunction with software using object or object-oriented software development environments that provide portable source code that can be used on a variety of computer or workstation platforms. Alternatively, the disclosed system may be implemented partially or fully in hardware using standard logic circuits or VLSI design. Whether software or hardware is used to implement the systems in accordance with this disclosure is dependent on the speed and/or efficiency requirements of the system, the particular function, and the particular software or hardware systems or microprocessor or microcomputer systems being utilized.

[0119] In yet another embodiment, the disclosed methods may be partially implemented in software that can be stored on a storage medium, executed on programmed general-purpose computer with the cooperation of a controller and memory, a special purpose computer, a microprocessor, or the like. In these instances, the systems and methods of this disclosure can be implemented as program embedded on personal computer such as an applet, JAVA® or CGI script, as a resource residing on a server or computer workstation, as a routine embedded in a dedicated measurement system, system component, or the like. The system can also be implemented by physically incorporating the system and/or method into a software and/or hardware system.

[0120] Although the present disclosure describes components and functions implemented in the aspects, embodiments, and/or configurations with reference to particular standards and protocols, the aspects, embodiments, and/or configurations are not limited to such standards and

protocols. Other similar standards and protocols not mentioned herein are in existence and are considered to be included in the present disclosure. Moreover, the standards and protocols mentioned herein and other similar standards and protocols not mentioned herein are periodically superseded by faster or more effective equivalents having essentially the same functions. Such replacement standards and protocols having the same functions are considered equivalents included in the present disclosure.

[0121] The present disclosure, in various aspects, embodiments, and/or configurations, includes components, methods, processes, systems and/or apparatus substantially as depicted and described herein, including various aspects, embodiments, configurations embodiments, subcombinations, and/or subsets thereof. Those of skill in the art will understand how to make and use the disclosed aspects, embodiments, and/or configurations afrer understanding the present disclosure. The present disclosure, in various aspects, embodiments, and/or configurations, includes providing devices and processes in the absence of items not depicted and/or described herein or in various aspects, embodiments, and/or configurations hereof, including in the absence of such items as may have been used in previous devices or processes, e.g., for improving performance, achieving ease and\or reducing cost of implementation.

[0122] The foregoing discussion has been presented for purposes of illustration and description. The foregoing is not intended to limit the disclosure to the form or forms disclosed herein. In the foregoing Detailed Description for example, various features of the disclosure are grouped together in one or more aspects, embodiments, and/or configurations for the purpose of streamlining the disclosure. The features of the aspects, embodiments, and/or configurations of the disclosure may be combined in alternate aspects, embodiments, and/or configurations other than those discussed above. This method of disclosure is not to be interpreted as reflecting an intention that the claims require more features than are expressly recited in each claim. Rather, as the following claims reflect, inventive aspects lie in less than all features of a single foregoing disclosed aspect, embodiment, and/or configuration. Thus, the following claims are hereby incorporated into this Detailed Description, with each claim standing on its own as a separate preferred embodiment of the disclosure.

[0123] Moreover, though the description has included description of one or more aspects, embodiments, and/or configurations and certain variations and modifications, other variations, combinations, and modifications are within the scope of the disclosure, e.g., as may be within the

skill and knowledge of those in the art, afrer understanding the present disclosure. It is intended to obtain rights which include alternative aspects, embodiments, and/or configurations to the extent permitted, including alternate, interchangeable and/or equivalent structures, functions, ranges or steps to those claimed, whether or not such alternate, interchangeable and/or equivalent structures, functions, ranges or steps are disclosed herein, and without intending to publicly dedicate any patentable subject matter.

What is claimed is:

1. A method, comprising:
providing a device having first and second screens;
receiving an instruction to launch a first application;
determining whether the first application is set to receive a keyboard focus upon launch;
one of:
presenting a virtual keyboard after determining that the first application is set to receive a keyboard focus upon launch; and
not presenting a virtual keyboard after determining that the first application is not set to receive a keyboard focus upon launch.

2. The method of claim 1, wherein it is determined that the first application is set to receive a keyboard focus upon launch, and wherein the virtual keyboard is presented by at least one of the first and second screens.

3. The method of claim 2, wherein at the time the instruction to launch the first application was received the virtual keyboard was displayed by at least one of the first and second screens, and wherein after the first application is launched the virtual keyboard continues to be displayed.

4. The method of claim 3, wherein at the time the instruction to launch the first application was received a second application was operable and the virtual keyboard displayed at that time was operable to receive input in connection with the second application, and wherein after the first application is launched the virtual keyboard continues to be displayed and is operable to receive input in connection with the first application.

5. The method of claim 1, wherein it is determined that the first application is not set to receive a keyboard focus upon launch, and wherein a virtual keyboard is not presented by at least one of the first and second screens after the first application is launched.

6. The method of claim 5, wherein at the time the instruction to launch the first application was received a virtual keyboard was displayed by at least one of the first and second screens, and wherein after the first application is launched the virtual keyboard is no longer displayed.

7. The method of claim 1, wherein at the time the instruction to launch the first application is received no virtual keyboard is presented by the device, wherein it is determined that the first application is set to receive a keyboard focus upon launch, and wherein after it is determined that the first application is set to receive a keyboard focus upon launch the virtual keyboard is presented by at least one of the first and second screens.

8. The method of claim 1, wherein the instruction to launch the first application is received ftom a user.

9. The method of claim 1, wherein the instruction to launch the first application is received as a result of the execution of a second application.

10. The method of claim 1, wherein a first portion of the virtual keyboard is presented using a first portion of the first screen, wherein a second portion of the virtual keyboard is presented using a first portion of the second screen, wherein first information is presented using a second portion of the first screen, and wherein second information is presented using a second portion of the second screen;

11. A device, comprising:
at least a first screen, the first screen including a touch sensitive display area;
memory;
a processor;
application programming stored in the memory and executed by the processor, wherein the application programming is operable to:
receive an instruction to launch a first application;
determine whether the first application is to have keyboard focus on launch of the first application;
present a virtual keyboard in response to determining that the first application is to have keyboard focus of a user on launch of the first application.

12. The device of claim 11, wherein the application programming is further operable to present the virtual keyboard in connection with a second application that is operable at the time the instruction to launch the first application is received.

13. The device of claim 12, wherein in response to determining that the first application is to have keyboard focus on launch of the first application, the application programming is further operable to continue to present the virtual keyboard on launch of the first application.

14. The device of claim 12, wherein in response to determining that the first application is not to have keyboard focus on launch of the first application, the application programming is further operable to discontinue the presentation of the virtual keyboard on the launch of the first application.

15. The device of claim 11, further comprising a second screen, wherein the second screen includes a touch sensitive area, wherein in response to determining that the first application is to have keyboard focus on launch of the first application, the application programming is further operable to present the virtual keyboard on launch of the first application on at least a portion of the first screen and on at least a portion of a second screen.

16. A computer readable medium having stored thereon computer-executable instructions, the computer executable instructions causing a processor to execute a method for providing a virtual keyboard, the computer-executable instructions comprising:
instructions to receive a command to launch a first application;
instructions to determine whether the first application is to receive a keyboard focus on launch of the first application;
at least one of:
instructions to display a virtual keyboard on launch in response to determining that the first application is to receive a keyboard focus on launch of the first application;
instructions to not display a virtual keyboard on launch in response to determining that the first application is not to receive a keyboard focus on launch of the first application.

17. The computer readable medium of claim 16, further comprising:
instructions to discontinue a presentation of a virtual keyboard presented in connection with a second application at the time the command to launch the first application is received in response to determining that the first application is not to receive a keyboard focus on launch of the first application.

18. The computer readable medium of claim 16, further comprising:
instructions to continue the presentation of a virtual keyboard presented in connection with a second application at the time the command to launch the first application is received in response to determining that the first application is to receive a keyboard focus on launch of the first application.

19. The computer readable medium of claim 17, further comprising:
instructions to display a virtual keyboard on launch in response to determining that the first application is to receive a keyboard focus on launch of the first application.

20. The computer readable medium of claim 17, further comprising: instructions to not display a virtual keyboard on launch in response to determining that the first application is not to receive a keyboard focus on launch of the first application.

Attorney Docket No. 6583-60

ABSTRACT

Methods and devices for selectively presenting a virtual keyboard are provided. More particularly, upon the receipt of instructions to launch an application, a determination can be made as to whether the application is associated with instructions to receive keyboard focus on launch. If the application is to receive keyboard focus on launch, a virtual keyboard is presented together with the newly launched application. Where an application is not set to receive keyboard focus on launch, a virtual keyboard that is presented when instructions to launch the application are received can be dismissed.

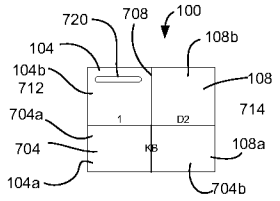


Fig. 7A

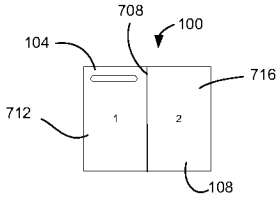


Fig. 7B

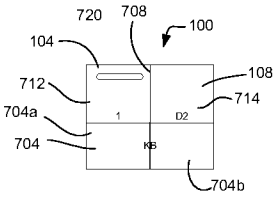


Fig. 8A

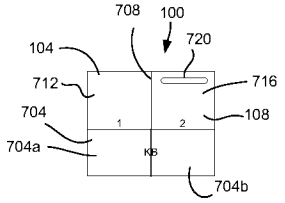


Fig. 8B

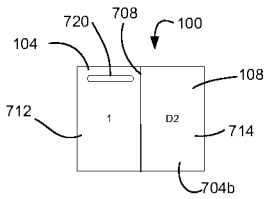


Fig. 9A

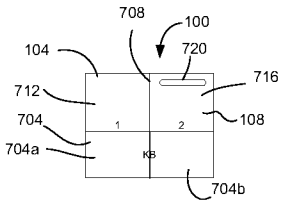


Fig. 9B

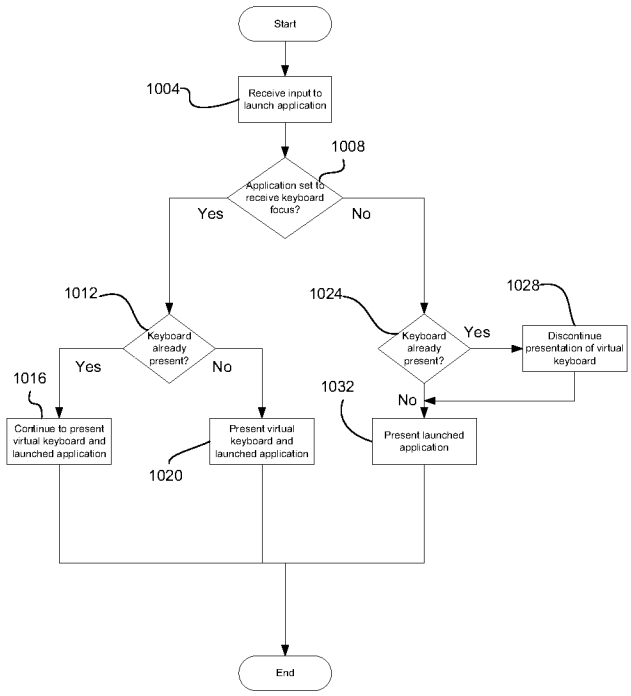


Fig. 10

KEYBOARD MAXIMIZATION

[0100] Fig. 7A depicts a device 100 with first 104 and second 108 screens in an open state, with both screens 104 and 108 in view of the user. As described elsewhere herein, the first 104 and second 108 screens can comprise touch sensitive screens. In this first scenario, a virtual keyboard 704 is presented by the screens 104, 108. In particular, a first portion 704a of the virtual keyboard 704 is presented by a first portion 104a of the first screen 104, while a second portion 704b of the virtual keyboard 704 is presented by a first portion 108a of the second screen 108. Accordingly, in this dual portrait orientation, the virtual keyboard 704 spans a line of separation 708 between the first 104 and second 108 screens. Moreover, the virtual keyboard 704 does not occupy all of the area of either the first screen 104 or the second screen 108. Accordingly, a second portion 104b of the first screen 104 and a second portion 108b of the second screen 108 remain available for the presentation of information other than the virtual keyboard 704 to a user. For instance, as shown in this example, at least a portion of a first application or window 712 is displayed by a second portion 104b of the first screen 104, while a second application or window 716 is displayed by a second portion 108b of the second screen 108. In accordance with embodiments of the present invention, the second application or window 716 can be a second page or an extension of the first application or window 712, or the second application or window 716 can comprise a separate application. In this example scenario, the keyboard focus is on the first application 712. This can be indicated by the provision of an input field 720. As used herein, keyboard focus means that a virtual keyboard 704 or other input means is operable with respect to the application 712 or 716 having the keyboard focus, allowing a user to provide input to the application 712 or 716 or that affects the application 712 or 716 through the virtual keyboard 704 or other input means. In accordance with other embodiments, the second screen 108 can present a desktop or can be blank. In accordance with still other embodiments, the second screen 108 can present a second page of the first application 712, or can otherwise provide an expanded view of the first application 712, for example where the first application 712 is a dual or multi-screen application.

[0101] The configuration or operating mode of the device 100 illustrated in Fig. 7A is a multiple screen (e.g., dual screen) operating mode, in which both the first screen 104 and the second screen 108 are in view of the user. Moreover, the first screen 104 and the second screen 108 can both be operable to present information to the user. As described elsewhere herein,

to the multiple screen or open operating mode when the screens 104 and 108 are in a dual portrait orientation.

[0103] Fig. 8A depicts a device 100 on which a virtual keyboard 704 is presented by portions of first 104 and second 108 screens, while the device 100 is in the open operating mode. In this example scenario, the keyboard focus is on the second application or window 716. The keyboard focus is represented by the provision of an input field 720 in an area of the second screen 108 presenting at least a portion of the second application or window 716. In this multiple screen or open operating mode, and with the device 100 oriented such that the first 104 and second 108 screens are in a dual portrait orientation, a virtual keyboard 704 spans the line of separation 708 between the first 104 and second 108 screens.

[0104] In Fig. 8B, the device 100 of Fig. 8A is shown, after the user has changed the operating mode of the device 100 to the closed operating mode with a single, primary screen 104 in view. Moreover, in this example, the first screen 104 presents the first application or window 712, but does not present a virtual keyboard. More particularly, in this example, the user input, in the form of closing the device, causes the virtual keyboard 704 that had been presented while the device 100 was in the multiple screen operating mode to be dismissed, because the application that was on the primary screen 104, here the first application 712, did not have keyboard focus. More particularly, the application that did have keyboard focus, the second application 716, is now out of view. In addition, by dismissing the virtual keyboard in this instance, the size of the first window 712 can be increased, such that the first window 712 occupies all of the first screen 104.

[0105] Fig. 8C depicts the device 100 of Fig. 8B, after that device 100 has been returned to the multiple screen operating mode, with both the first 104 and second 108 screens in the view of the user. Here, opening the device 100 does not recall the virtual keyboard, since the application that had keyboard focus (the second application 716) prior to closing the device does not retain keyboard focus automatically when the device 100 is reopened.

[0106] Fig. 9A depicts a multiple screen device 100 in another exemplary scenario. In this scenario, the device 100 is depicted in the multiple screen operating mode, with the first 104 and second 108 screens in a dual landscape orientation. Moreover, in this example, a virtual keyboard 704 is presented using all or substantially all of the second screen 108, while a first window 712 is presented using all or substantially all of the first screen 104. Moreover, in this

embodiments of the present invention provide first 104 and second 108 screens that are in a hinged relationship to one another, such that the device 100 can be opened or closed, thereby changing the operating mode of the device 100. Alternatively, the first 104 and second 108 screens can comprise virtual screens.

[0102] In Fig. 7B, the operating mode of the device 100 has been changed from the operating mode shown in Fig. 7A. In particular, the screens 104, 108 have been folded around the hinged interconnection, closing the device 100 and leaving the first, primary screen 104 in view of the user and operative. Moreover, the second screen 108 can be placed in an inoperative or idle state in which it displays no information. With only one screen 104 in view of the user, the available display area in this single screen or closed operating mode is diminished as compared to the open or multiple screen operating mode illustrated in Fig. 7A. In response to this change in the operating mode, the information displayed to the user can be revised. In particular, the first screen 104 in this example continues to present a virtual keyboard 704 and at least a portion of the first application 712 to the user. In addition, the input field 720 continues to be presented. In this example, closing the device 100 retains the virtual keyboard 704 in view, because the application presented by the primary or first screen 104, the first application 712, remains in view and was the application with keyboard focus when the device 100 was placed in the closed or single screen operating mode. In accordance with embodiments of the present invention, the format of the virtual keyboard 704 can be compressed while the device 100 is in the closed operating mode, as compared to the format of the virtual keyboard 704 when the device 100 is in the open operating mode. In accordance with further embodiments of the present invention, the input options provided by a virtual keyboard 704 in the different operating modes can remain constant. Alternatively, fewer input options or virtual keys can be presented by the virtual keyboard 704 when the device 100 is in the closed operating mode. From the configuration illustrated in Fig. 7B, if the user opens the device 100, to place it in the first operating mode, the second screen 108 is placed back into the view of the user and becomes operative, such that the virtual keyboard 704 is displayed across portions of the first 104 and second screens 108, as shown in Fig. 7A. In addition, the window or information displayed by the second portion 108b of the second screen 108 can be the same or different than the information that was displayed in the second window 716 in the configuration illustrated in Fig. 7A. Accordingly, the size of the virtual keyboard 704 can be adjusted, such that it is maximized when the device 100 is returned

example the first application 712 has keyboard focus. The placement of keyboard focus on the first application 712 is indicated by the provision of an input field 720.

[0107] In Fig. 9B, the device 100 of Fig. 9A is shown, after the user has placed the device 100 in the single screen operating mode, with only the primary or first screen 104 in view of the user. With the device in the single screen operating mode, in this example a portion of the first application or window 712 is presented by a portion of the first screen 104, while a second portion of the first screen 104 is used to present the virtual keyboard 704. In addition, an input field 720 can continue to be presented. In this example, closing the device retains the virtual keyboard 704 in view, because the first application 712 remains in view, and continues to have keyboard focus. Accordingly, a presentation of a virtual keyboard 704 can be continued, even when the device 100 is closed and the primary screen 104 is in a landscape orientation. In addition, in response to the user opening the device 100 while configured as illustrated in Fig. 9B, the presentation by the screens 104 and 108 will return to the format illustrated in Fig. 9A. Accordingly, the size of the virtual keyboard 704 can be adjusted, such that it is maximized when the device 100 is returned to the multiple screen operating mode when the screens 104 and 108 are in a dual landscape orientation.

[0108] Fig. 10A depicts the output of a device 100 in a multiple screen operating mode, with the first 104 and second 108 screens in view of the user. In this example, a virtual keyboard 704 is presented by all or substantially all of the second screen 108, while a first application or window 712 is presented by all or substantially all of the first screen 104. Notably, the first application 712 in this example does not have keyboard focus.

[0109] In Fig. 10B, the output of the device 100 of Fig. 10A is depicted, after the user has closed the device 100, thereby changing the operating mode of the device to the single screen operating mode in which only the first or primary screen 104 is in the view of the user. As shown in the figure, in this example the first application window 712 is presented, but the virtual keyboard 704 is not presented. Here, closing the device has the effect of dismissing the keyboard since the window or application presented by the first or primary screen 104 did not have keyboard focus. From the state depicted in Fig. 10B, returning the device 100 to the multiple screen operating mode would not bring up the virtual keyboard.

[0110] Fig. 11 depicts aspects of a method for selectively providing a virtual keyboard 704 and maximizing the area of the keyboard 704 in accordance with embodiments of the present

invention. Initially, at step 1104, a virtual keyboard 704 and a first window or application 712 is presented to a user. This initial presentation can be provided while the device 100 is in either of the multiple screen or single screen operating modes. Moreover, a second window or application 716 can also be displayed, for example when the device 100 is in the multiple screen operating mode. However, even in the multiple screen operating mode, the display of a second window or application 716 is not required. For example, the second screen 108 can display a desktop or can be blank.

[0111] At step 1108, input is received from the user changing the operating mode of the device 100. For example, the input can close the device, thereby changing the operating mode from the multiple screen or open operating mode, in which both screens 104 and 108 are in view of the user, to the single screen or closed operating mode, in which a single screen, i.e., the first or primary screen 104, is in view of the user. Alternatively, the user input can change the device 100 from the single screen operating mode to the multiple screen operating mode. A signal indicating a change in the operating mode can be provided by one or more position sensors 172. Alternatively or in addition, one or more position sensors 172 can indicate whether the current state of the device 100 is the multiple screen operating mode or the single screen operating mode.

[0112] At step 1112, a determination can be made as to whether the change in the operating mode of the device 100 retained the screen 104 or 108 having a window or application 712 or 716 with keyboard focus in view. If the change in operating mode caused the screen 104 or 108 presenting the application or window 712 or 716 to be removed from the view of the user, the virtual keyboard 704 is dismissed, and the application or window 712 or 716 that did not have keyboard focus is expanded (step 1116). More particularly, since a previously displayed application 712 or 716 has been removed from the view of the user, this implies that the device 100 has been placed in the single screen operating mode, with only the primary or first screen 104 in view of the user. Accordingly, expanding the application or window 712 or 716 that did not have keyboard focus (and that continues to not have keyboard focus) means expanding that application or window 712 or 716 to fill all or substantially all of the first or primary screen 104. In accordance with embodiments of the present invention, placing the device 100 in the single screen operating mode can result in the second screen 108 being powered off or idled, at least while the device 100 remains in the single screen operating mode.

[0113] If it is determined that the screen presenting the application or window 712 or 716 remains in view afrer the change in operating mode, this implies that the change was from the single screen operating mode to the multiple screen operating mode and the virtual keyboard 704 continues to be displayed (step 1120). In addition, the virtual keyboard 704 can be expanded. For instance, where the device 100 is oriented such that in the multiple screen operating mode the screens 104 and 108 are in a dual portrait format, the virtual keyboard 704 is expanded to extend across portions of the first screen and the second screen 108. Where the device 100 is oriented such that in the first operating mode the screens 104 and 108 are in a dual landscape format, the virtual keyboard 704 is expanded to extend across all or most of the screen 104 or 108 comprising a bottom screen, as determined, for example, by an accelerometer 176.

[0114] Afrer dismissing the presentation of the virtual keyboard at step 1116, or continuing to display the virtual keyboard 1120, the process for selectively presenting a keyboard can end. As can be appreciated by one of skill in the art afrer consideration of the present disclosure, this process can be performed whenever there is a change in the operating mode of the device 100 while a virtual keyboard 704 is being presented. Moreover, embodiments of the present invention allow for the size of the virtual keyboard 704 to be maximized, in order to present a virtual keyboard 704 that maximizes the ease of use of the virtual keyboard 704, depending on the operating mode. In particular, embodiments of the present invention allow the virtual keyboard 704 to be expanded while the device 100 is in the multiple screen operating mode, and to either continue to be displayed or to be dismissed while the device is in the single screen operating mode.

[0115] Although various examples and illustrations have discussed operation of a device 100 having first 104 and second 108 screens in which the first screen 104 is considered a primary screen, the screen 104 or 108 that is treated as primary can be varied, depending on the context in which the device 100 is used and/or oriented. For example, the second screen 108 may operate as the primary screen where the device 100 is oriented in a multiple screen operating mode such that the second screen 108 is to the lefr and the first screen 104 is to the right, when the screens 104 and 108 are in view of the user.

[0116] In accordance with embodiments of the present invention, the method disclosed herein can be performed by the execution of application programming or other software components stored in memory 208, 508 by a processor 204, 504. For instance, the operating system 516, a

component or components of the framework 520, and/or a component or components of the multi-display management module 524 can include or provide functionality to receive and act on input received from a user in connection with the display and operation of a virtual keyboard 704. Moreover, such input can include input comprising user interface interactions by a user that are managed by the input management module 536, or other components. In accordance with embodiments of the present invention, the user input can further include moving the screens 104 and 108 relative to one another, between open and closed operating states. Detection of a current operating state and/or a change in operating state can be provided by one or more position sensors 172. In addition, although certain embodiments have been described in connection with operation on a device having first 104 and second 108 screens, the invention is not limited to operation on such a device. For example, a device 100 can have more than two screens, and/or the screen can comprise virtual screens or windows.

[0117] The exemplary systems and methods of this disclosure have been described in relation to the selective presentation of a virtual keyboard 704 in response to a detected operating mode of the device 100 and whether an application having keyboard focus is in view of the user. However, to avoid unnecessarily obscuring the present disclosure, the preceding description omits a number of known structures and devices. This omission is not to be construed as a limitation of the scopes of the claims. Specific details are set forth to provide an understanding of the present disclosure. It should however be appreciated that the present disclosure may be practiced in a variety of ways beyond the specific detail set forth herein.

[0118] Furthermore, while the exemplary aspects, embodiments, and/or configurations illustrated herein show the various components of the system collocated, certain components of the system can be located remotely, at distant portions of a distributed network, such as a LAN and/or the Internet, or within a dedicated system. Thus, it should be appreciated, that the components of the system can be combined in to one or more devices, such as a computer, laptop, netbook, tablet computer, smart phone, mobile device, etc., or collocated on a particular node of a distributed network, such as an analog and/or digital telecommunications network, a packet-switch network, or a circuit-switched network. It will be appreciated from the preceding description, and for reasons of computational efficiency, that the components of the system can be arranged at any location within a distributed network of components without affecting the operation of the system. For example, the various components can be located in a switch such as

a PBX and media server, gateway, in one or more communications devices, at one or more users' premises, or some combination thereof. Similarly, one or more functional portions of the system could be distributed between a telecommunications device(s) and an associated computing device.

[0119] Furthermore, it should be appreciated that the various links connecting the elements can be wired or wireless links, or any combination thereof, or any other known or later developed element(s) that is capable of supplying and/or communicating data to and from the connected elements. These wired or wireless links can also be secure links and may be capable of communicating encrypted information. Transmission media used as links, for example, can be any suitable carrier for electrical signals, including coaxial cables, copper wire and fiber optics, and may take the form of acoustic or light waves, such as those generated during radio-wave and infra-red data communications.

[0120] Also, while the flowcharts have been discussed and illustrated in relation to a particular sequence of events, it should be appreciated that changes, additions, and omissions to this sequence can occur without materially affecting the operation of the disclosed embodiments, configuration, and aspects.

[0121] A number of variations and modifications of the disclosure can be used. It would be possible to provide for some features of the disclosure without providing others.

[0122] For example in one alternative embodiment, while a virtual keyboard 704 is displayed, a first 712 and/or second 716 window can be displayed in a reduced format or size, as compared to when a virtual keyboard 704 is not presented to the user.

[0123] In another alternative embodiment, while a virtual keyboard 704 is displayed, a first 712 and/or second 716 window can be displayed in a reduced format or size, as compared to when a virtual keyboard 704 is not presented to the user.

[0124] In accordance with still other embodiments, the screens 104, 108 may comprise virtual screens or display areas, and selecting different operating modes can include selecting between dual and single virtual screen modes.

[0125] In yet another embodiment, the systems and methods of this disclosure can be implemented in conjunction with a special purpose computer, a programmed microprocessor or microcontroller and peripheral integrated circuit element(s), an ASIC or other integrated circuit, a digital signal processor, a hard-wired electronic or logic circuit such as discrete element circuit,

a programmable logic device or gate array such as PLD, PLA, FPGA, PAL, special purpose computer, any comparable means, or the like. In general, any device(s) or means capable of implementing the methodology illustrated herein can be used to implement the various aspects of this disclosure. Exemplary hardware that can be used for the disclosed embodiments, configurations and aspects includes computers, handheld devices, telephones (e.g., cellular, Internet enabled, digital, analog, hybrids, and others), and other hardware known in the art. Some of these devices include processors (e.g., a single or multiple microprocessors), memory, nonvolatile storage, input devices, and output devices. Furthermore, alternative software implementations including, but not limited to, distributed processing or component/object distributed processing, parallel processing, or virtual machine processing can also be constructed to implement the methods described herein.

[0126] In yet another embodiment, the disclosed methods may be readily implemented in conjunction with software using object or object-oriented software development environments that provide portable source code that can be used on a variety of computer or workstation platforms. Alternatively, the disclosed system may be implemented partially or fully in hardware using standard logic circuits or VLSI design. Whether software or hardware is used to implement the systems in accordance with this disclosure is dependent on the speed and/or efficiency requirements of the system, the particular function, and the particular software or hardware systems or microprocessor or microcomputer systems being utilized.

[0127] In yet another embodiment, the disclosed methods may be partially implemented in software that can be stored on a storage medium, executed on programmed general-purpose computer with the cooperation of a controller and memory, a special purpose computer, a microprocessor, or the like. In these instances, the systems and methods of this disclosure can be implemented as program embedded on personal computer such as an applet, JAVA® or CGI script, as a resource residing on a server or computer workstation, as a routine embedded in a dedicated measurement system, system component, or the like. The system can also be implemented by physically incorporating the system and/or method into a software and/or hardware system.

[0128] Although the present disclosure describes components and functions implemented in the aspects, embodiments, and/or configurations with reference to particular standards and protocols, the aspects, embodiments, and/or configurations are not limited to such standards and

protocols. Other similar standards and protocols not mentioned herein are in existence and are considered to be included in the present disclosure. Moreover, the standards and protocols mentioned herein and other similar standards and protocols not mentioned herein are periodically superseded by faster or more effective equivalents having essentially the same functions. Such replacement standards and protocols having the same functions are considered equivalents included in the present disclosure.

[0129] The present disclosure, in various aspects, embodiments, and/or configurations, includes components, methods, processes, systems and/or apparatus substantially as depicted and described herein, including various aspects, embodiments, configurations embodiments, subcombinations, and/or subsets thereof. Those of skill in the art will understand how to make and use the disclosed aspects, embodiments, and/or configurations after understanding the present disclosure. The present disclosure, in various aspects, embodiments, and/or configurations, includes providing devices and processes in the absence of items not depicted and/or described herein or in various aspects, embodiments, and/or configurations hereof, including in the absence of such items as may have been used in previous devices or processes, e.g., for improving performance, achieving ease and\or reducing cost of implementation.

[0130] The foregoing discussion has been presented for purposes of illustration and description. The foregoing is not intended to limit the disclosure to the form or forms disclosed herein. In the foregoing Detailed Description for example, various features of the disclosure are grouped together in one or more aspects, embodiments, and/or configurations for the purpose of streamlining the disclosure. The features of the aspects, embodiments, and/or configurations of the disclosure may be combined in alternate aspects, embodiments, and/or configurations other than those discussed above. This method of disclosure is not to be interpreted as reflecting an intention that the claims require more features than are expressly recited in each claim. Rather, as the following claims reflect, inventive aspects lie in less than all features of a single foregoing disclosed aspect, embodiment, and/or configuration. Thus, the following claims are hereby incorporated into this Detailed Description, with each claim standing on its own as a separate preferred embodiment of the disclosure.

[0131] Moreover, though the description has included description of one or more aspects, embodiments, and/or configurations and certain variations and modifications, other variations, combinations, and modifications are within the scope of the disclosure, e.g., as may be within the

skill and knowledge of those in the art, after understanding the present disclosure. It is intended to obtain rights which include alternative aspects, embodiments, and/or configurations to the extent permitted, including alternate, interchangeable and/or equivalent structures, functions, ranges or steps to those claimed, whether or not such alternate, interchangeable and/or equivalent structures, functions, ranges or steps are disclosed herein, and without intending to publicly dedicate any patentable subject matter.

What is claimed is:

1. A method, comprising:

providing a device having first and second screens;

in a first operating mode, presenting a virtual keyboard and a first window having a keyboard focus, wherein the virtual keyboard has a first keyboard size and format, and wherein the first window has a first window size;

receiving input ftom a user, wherein the input includes a change in the number of screens that are in view of the user and places the device in a second operating mode;

determining that the first window having keyboard focus remains in view of the user in the second operating mode;

in response to the change in the number of screen that are in view of the user and the determination that the first window having keyboard focus remains in view of the user in the second operating mode, presenting the first window and a virtual keyboard, wherein the virtual keyboard has a second keyboard size and format, and wherein the first window has one of the first window size and a second window size.

2. The method of claim 1, wherein in the first operating mode the first and second screens are in view of the user, wherein a first portion of the virtual keyboard is presented using a first portion of the first screen, wherein a second portion of the virtual keyboard is presented using a first portion of the second screen, and wherein the first window is presented using one of a second portion of the first screen and a second portion of the second screen.

3. The method of claim 2, wherein in the second operating mode only one of the first screen and the second screen is in view of the user, and wherein the first window and a virtual keyboard are presented using the one of the first screen and the second screen that is in the view of the user.

4. The method of claim 3, wherein the virtual keyboard displayed in the first operating mode is larger than the virtual keyboard displayed in the second operating mode.

5. The method of claim 4, wherein the virtual keyboard displayed in the first operating mode and the virtual keyboard displayed in the second operating mode present the same input options to the user.

6. The method of claim 3, wherein in the second operating mode the first window has the first window size.

7. The method of claim 3, wherein in the second operating mode the first window has the second window size, wherein the second window size is different than the first window size and format.

8. The method of claim 3, wherein the input received from the user includes closing the device.

9. The method of claim 1, wherein in the first operating mode only one of the first and second screens is in view of the user.

10. The method of claim 9, wherein in the second operating mode both the first and second screens are in view of the user.

11. A device, comprising:

a first screen, the first screen including a touch sensitive display area;

a second screen, the second screen including a touch sensitive display area, wherein in a first operating mode both the first screen and the second screen are in view of the user, and wherein in a second operating mode only one of the first and second screens is in view of the user;

memory;

a processor;

application programming stored in the memory and executed by the processor, wherein the application programming is operable to:

present a virtual keyboard and at least a portion of a first window having a keyboard focus, wherein the virtual keyboard is presented in a first keyboard size and format, and wherein the first window is presented in a first window size;

while presenting the virtual keyboard and at least a first portion of a first window having the keyboard focus, detect a change in a current operating mode;

determine whether the at least a portion of the first window having the keyboard focus remains in view of the user after the detected change in operating mode;

in response to detecting the change in the current operating mode and to determining that the first window having the keyboard focus remains in view of the user after the change in operating mode, presenting at least one of the first portion and a second portion of the first window and the virtual keyboard, wherein the virtual keyboard is presented in a second keyboard size and format, and wherein the first window is presented in one of the first window size and a second window size.

12. The device of claim 11, further comprising:

a hinge, wherein the hinge interconnects the first and second screens, wherein in the first operating mode the device is in an open state, and wherein in the second operating mode the device is in a closed state.

13. The device of claim 12, wherein in the first operating mode the virtual keyboard is presented by a first portion of the first screen and a first portion of the second screen, and

wherein the at least a first portion of the first window is presented by a second portion of the first screen.

14. The device of claim 13, wherein the first screen is a primary screen.

15. The device of claim 13, wherein the first window has a keyboard focus while the device is in the first operating mode and while the device is in the second operating mode.

16. A computer readable medium having stored thereon computer-executable instructions, the computer executable instructions causing a processor to execute a method for providing a virtual keyboard, the computer-executable instructions comprising:

instructions to determine an operating mode of a device having at least a first touch sensitive screen and a second touch sensitive screen;

instructions to display a virtual keyboard and at least a first portion of a first window having a keyboard focus in a first operating mode, wherein the virtual keyboard has a first keyboard size and format in the first operating mode;

instructions to detect a change ftom the first operating mode to a second operating mode and to determine whether the at least a first portion of the first window having the keyboard focus continues to be displayed, and in response to the detected change in operating mode and to a determination that the at least a first portion of the first window having the keyboard focus continues to be displayed, present the virtual keyboard and at least one of the first portion of the first window and a second portion of the first window, wherein the virtual keyboard has a second keyboard size and format in the second operating mode.

17. The computer readable medium of claim 16, the computer-executable instructions further comprising:

instructions to identify the first window as a window having a keyboard focus.

18. The computer readable medium of claim 17, wherein in the first operating mode both of the first and second touch sensitive screens are operative, and wherein in the second operating mode only one of the first and second touch sensitive screens is operative.

19. The computer readable medium of claim 18, wherein in the first operating mode the virtual keyboard is presented using a first portion of the first touch sensitive screen and a first portion of the second touch sensitive screen, wherein the first portion of the first window is presented by a second portion of the first touch sensitive screen, and wherein at least a first portion of a second window is presented by a second portion of the second touch sensitive screen.

20. The computer readable medium of claim 16, wherein in the first operating mode only one of the first and second touch sensitive screens is operative, and wherein in the second operating mode both of the first and second touch sensitive screens are operative.

Attorney Docket No. 6583-61

ABSTRACT

Methods and devices for selectively presenting a virtual keyboard are provided. More particularly, a change in the operating mode of a multiple screen device from a multiple screen operating mode to a single screen operating mode, or from a single screen operating mode to a multiple screen operating mode, can be determined. Moreover, a change in the operating mode can effect a change in a presentation of a virtual keyboard. More particularly, a presentation of a virtual keyboard can be retained, where the number of screens of the device in view of the user is changed, provided an application with a keyboard focus remains in view of the user after the change in operating mode.

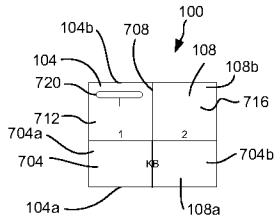


Fig. 7A

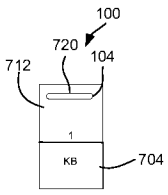


Fig. 7B

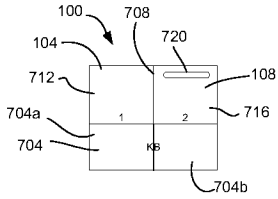


Fig. 8A

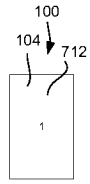


Fig. 8B

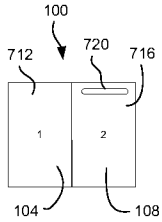


Fig. 8C

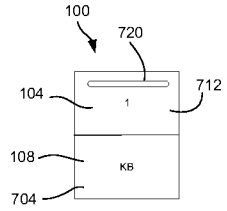


Fig. 9A

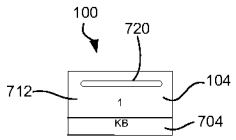


Fig. 9B

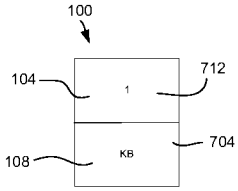


Fig. 10A

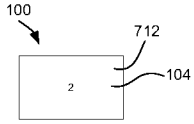


Fig. 10B

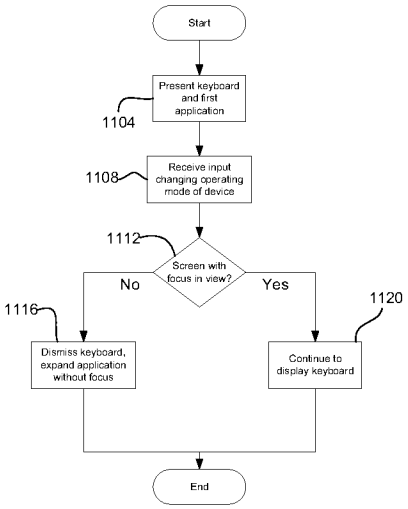


Fig. 11

KEYBOARD DISMISSED ON CLOSURE OF DEVICE

[0100] Fig. 7A depicts a device 100 with first 104 and second 108 screens in an open state, with both screens 104 and 108 in view of the user. As described elsewhere herein, the first 104 and second 108 screens can comprise touch sensitive screens. In this first scenario, a virtual keyboard 704 is presented by the screens 104, 108. In particular, a first portion 704a of the virtual keyboard 704 is presented by a first portion 104a of the first screen 104, while a second portion 704b of the virtual keyboard 704 is presented by a first portion 108a of the second screen 108. Accordingly, in this dual portrait orientation, the virtual keyboard 704 spans a line of separation 708 between the first 104 and second 108 screens. Moreover, the virtual keyboard 704 does not occupy all of the area of either the first screen 104 or the second screen 108. Accordingly, a second portion 104b of the first screen 104 and a second portion 108b of the second screen 108 remain available for the presentation of information other than the virtual keyboard 704 to a user. For instance, as shown in this example, at least a portion of a first application or window 712 is displayed by a second portion 104b of the first screen 104, while a second application or window 716 is displayed by a second portion 108b of the second screen 108. In accordance with embodiments of the present invention, the second application or window 716 can be a second page or an extension of the first application or window 712, or the second application or window 716 can comprise a separate application. In this example scenario, the keyboard focus is on the first application 712. This can be indicated by the provision of an input field 720. As used herein, keyboard focus means that a virtual keyboard 704 or other input means is operable with respect to the application 712 or 716 having the keyboard focus, allowing a user to provide input to the application 712 or 716 or that affects the application 712 or 716 through the virtual keyboard 704 or other input means. In accordance with other embodiments, the second screen 108 can present a desktop or can be blank. In accordance with still other embodiments, the second screen 108 can present a second page of the first application 712, or can otherwise provide an expanded view of the first application 712, for example where the first application 712 is a dual or multi-screen application.

[0101] The configuration or operating mode of the device 100 illustrated in Fig. 7A is a multiple screen or open operating mode, in which both the first screen 104 and the second screen 108 are in view of the user. Moreover, the first screen 104 and the second screen 108 can both be operable to present information to the user. As described elsewhere herein, embodiments of the present invention provide first 104 and second 108 screens that are in a hinged relationship to one another, such that the device 100 can be opened or closed, thereby changing the operating mode of the device 100. Alternatively, the first 104 and second 108 screens can comprise virtual screens.

[0102] In Fig. 7B, the operating mode of the device 100 has been changed from the operating mode shown in Fig. 7A. In particular, the screens 104, 108 have been folded around the hinged interconnection, closing the device 100 and leaving the first, primary screen 104 in view of the user and operative. Moreover, the second screen 108 can be placed in an inoperative or idle state in which it displays no information. With only one screen 104 in view of the user, the available display area in this single screen or closed operating mode is diminished as compared to the open or multiple screen operating mode illustrated in Fig. 7A. In response to this change in the operating mode, the information displayed to the user can be revised. In particular, the first screen 104 in this example continues to present a virtual keyboard 704 and at least a portion of the first application 712 to the user. In addition, the input field 720 continues to be presented. In this example, closing the device 100 retains the virtual keyboard 704 in view, because the application presented by the primary or first screen 104, the first application 712, remains in view and was the application with keyboard focus when the device 100 was placed in the closed or single screen operating mode. In accordance with embodiments of the present invention, the format of the virtual keyboard 704 can be compressed while the device 100 is in the closed operating mode, as compared to the format of the virtual keyboard 704 when the device 100 is in the open operating mode. In accordance with further embodiments of the present invention, the input options provided by a virtual keyboard 704 in the different operating modes can remain constant. Alternatively, fewer input options or virtual keys can be presented by the virtual keyboard 704 when the device 100 is in the closed operating mode. From the configuration illustrated in Fig. 7B, if the user opens the device 100, to place it in the first operating mode, the second screen 108 is placed back into the view of the user and becomes operative, such that the virtual keyboard 704 is displayed across portions of the first 104 and second screens 108, as shown in Fig. 7A. In addition, the window or information displayed by the second portion 108b of the second screen 108 can be the same or different than the information that was displayed in the second window 716 in the configuration illustrated in Fig. 7A. Accordingly, the size of the virtual keyboard 704 can be adjusted, such that it is maximized when the device 100 is returned

to the multiple screen or open operating mode when the screens 104 and 108 are in a dual portrait orientation.

[0103] Fig. 8A depicts a device 100 on which a virtual keyboard 704 is presented by portions of first 104 and second 108 screens, while the device 100 is in the open operating mode. In this example scenario, the keyboard focus is on the second application or window 716. The keyboard focus is represented by the provision of an input field 720 in an area of the second screen 108 presenting at least a portion of the second application or window 716. In this multiple screen or open operating mode, and with the device 100 oriented such that the first 104 and second 108 screens are in a dual portrait orientation, a virtual keyboard 704 spans the line of separation 708 between the first 104 and second 108 screens.

[0104] In Fig. 8B, the device 100 of Fig. 8A is shown, after the user has changed the operating mode of the device 100 to the closed operating mode with a single, primary screen 104 in view. Moreover, in this example, the first screen 104 presents the first application or window 712, but does not present a virtual keyboard. More particularly, in this example, the user input, in the form of closing the device, causes the virtual keyboard 704 that had been presented while the device 100 was in the multiple screen operating mode to be dismissed, because the application that was on the primary screen 104, here the first application 712, did not have keyboard focus. More particularly, the application that did have keyboard focus, the second application 716, is now out of view. In addition, by dismissing the virtual keyboard in this instance, the size of the first window 712 can be increased, such that the first window 712 occupies all of the first screen 104.

[0105] Fig. 8C depicts the device 100 of Fig. 8B, after that device 100 has been returned to the multiple screen operating mode, with both the first 104 and second 108 screens in the view of the user. Here, opening the device 100 does not recall the virtual keyboard, since the application that had keyboard focus (the second application 716) prior to closing the device does not retain keyboard focus automatically when the device 100 is reopened.

[0106] Fig. 9A depicts a multiple screen device 100 in another exemplary scenario. In this scenario, the device 100 is depicted in the multiple screen operating mode, with the first 104 and second 108 screens in a dual landscape orientation. Moreover, in this example, a virtual keyboard 704 is presented using all or substantially all of the second screen 108, while a first window 712 is presented using all or substantially all of the first screen 104. Moreover, in this

example the first application 712 has keyboard focus. The placement of keyboard focus on the first application 712 is indicated by the provision of an input field 720.

[0107] In Fig. 9B, the device 100 of Fig. 9A is shown, after the user has placed the device 100 in the single screen operating mode, with only the primary or first screen 104 in view of the user. With the device in the single screen operating mode, in this example a portion of the first application or window 712 is presented by a portion of the first screen 104, while a second portion of the first screen 104 is used to present the virtual keyboard 704. In addition, an input field 720 can continue to be presented. In this example, closing the device retains the virtual keyboard 704 in view, because the first application 712 remains in view, and continues to have keyboard focus. Accordingly, a presentation of a virtual keyboard 704 can be continued, even when the device 100 is closed and the primary screen 104 is in a landscape orientation. In addition, in response to the user opening the device 100 while configured as illustrated in Fig. 9B, the presentation by the screens 104 and 108 will return to the format illustrated in Fig. 9A. Accordingly, the size of the virtual keyboard 704 can be adjusted, such that it is maximized when the device 100 is returned to the multiple screen operating mode when the screens 104 and 108 are in a dual landscape orientation.

[0108] Fig. 10A depicts the output of a device 100 in a multiple screen operating mode, with the first 104 and second 108 screens in view of the user. In this example, a virtual keyboard 704 is presented by all or substantially all of the second screen 108, while a first application or window 712 is presented by all or substantially all of the first screen 104. Notably, the first application 712 in this example does not have keyboard focus.

[0109] In Fig. 10B, the output of the device 100 of Fig. 10A is depicted, after the user has closed the device 100, thereby changing the operating mode of the device to the single screen operating mode in which only the first or primary screen 104 is in the view of the user. As shown in the figure, in this example the first application window 712 is presented, but the virtual keyboard 704 is not presented. Here, closing the device has the effect of dismissing the keyboard since the window or application presented by the first or primary screen 104 did not have keyboard focus. From the state depicted in Fig. 10B, returning the device 100 to the multiple screen operating mode would not bring up the virtual keyboard.

[0110] Fig. 11 depicts aspects of a method for selectively providing a virtual keyboard 704 and maximizing the area of the keyboard 704 in accordance with embodiments of the present

invention. Initially, at step 1104, a virtual keyboard 704 and a first window or application 712 is presented to a user. This initial presentation can be provided while the device 100 is in either of the multiple screen or single screen operating modes. Moreover, a second window or application 716 can also be displayed, for example when the device 100 is in the multiple screen operating mode. However, even in the multiple screen operating mode, the display of a second window or application 716 is not required. For example, the second screen 108 can display a desktop or can be blank.

[0111] At step 1108, input is received from the user changing the operating mode of the device 100. For example, the input can close the device, thereby changing the operating mode from the multiple screen or open operating mode, in which both screens 104 and 108 are in view of the user, to the single screen or closed operating mode, in which a single screen, i.e., the first or primary screen 104, is in view of the user. Alternatively, the user input can change the device 100 from the single screen operating mode to the multiple screen operating mode. A signal indicating a change in the operating mode can be provided by one or more position sensors 172. Alternatively or in addition, one or more position sensors 172 can indicate whether the current state of the device 100 is the multiple screen operating mode or the single screen operating mode.

[0112] At step 1112, a determination can be made as to whether the change in the operating mode of the device 100 retained the screen 104 or 108 having a window or application 712 or 716 with keyboard focus in view. If the change in operating mode caused the screen 104 or 108 presenting the application or window 712 or 716 to be removed from the view of the user, the virtual keyboard 704 is dismissed, and the application or window 712 or 716 that did not have keyboard focus is expanded (step 1116). More particularly, since a previously displayed application 712 or 716 has been removed from the view of the user, this implies that the device 100 has been placed in the single screen operating mode, with only the primary or first screen 104 in view of the user. Accordingly, expanding the application or window 712 or 716 that did not have keyboard focus (and that continues to not have keyboard focus) means expanding that application or window 712 or 716 to fill all or substantially all of the first or primary screen 104. In accordance with embodiments of the present invention, placing the device 100 in the single screen operating mode can result in the second screen 108 being powered off or idled, at least while the device 100 remains in the single screen operating mode.

[0113] If it is determined that the screen presenting the application or window 712 or 716 remains in view afrer the change in operating mode, this implies that the change was from the single screen operating mode to the multiple screen operating mode and the virtual keyboard 704 continues to be displayed (step 1120). In addition, the virtual keyboard 704 can be expanded. For instance, where the device 100 is oriented such that in the multiple screen operating mode the screens 104 and 108 are in a dual portrait format, the virtual keyboard 704 is expanded to extend across portions of the first screen and the second screen 108. Where the device 100 is oriented such that in the first operating mode the screens 104 and 108 are in a dual landscape format, the virtual keyboard 704 is expanded to extend across all or most of the screen 104 or 108 comprising a bottom screen, as determined, for example, by an accelerometer 176.

[0114] Afrer dismissing the presentation of the virtual keyboard at step 1116, or continuing to display the virtual keyboard 1120, the process for selectively presenting a keyboard can end. As can be appreciated by one of skill in the art afrer consideration of the present disclosure, this process can be performed whenever there is a change in the operating mode of the device 100 while a virtual keyboard 704 is being presented. Moreover, embodiments of the present invention allow for the size of the virtual keyboard 704 to be maximized, in order to present a virtual keyboard 704 that maximizes the ease of use of the virtual keyboard 704, depending on the operating mode. In particular, embodiments of the present invention allow the virtual keyboard 704 to be expanded while the device 100 is in the multiple screen operating mode, and to either continue to be displayed or to be dismissed while the device is in the single screen operating mode.

[0115] Although various examples and illustrations have discussed operation of a device 100 having first 104 and second 108 screens in which the first screen 104 is considered a primary screen, the screen 104 or 108 that is treated as primary can be varied, depending on the context in which the device 100 is used and/or oriented. For example, the second screen 108 may operate as the primary screen where the device 100 is oriented in a multiple screen operating mode such that the second screen 108 is to the lefr and the first screen 104 is to the right, when the screens 104 and 108 are in view of the user.

[0116] In accordance with embodiments of the present invention, the method disclosed herein can be performed by the execution of application programming or other software components stored in memory 208, 508 by a processor 204, 504. For instance, the operating system 516, a

component or components of the framework 520, and/or a component or components of the multi-display management module 524 can include or provide functionality to receive and act on input received from a user in connection with the display and operation of a virtual keyboard 704. Moreover, such input can include input comprising user interface interactions by a user that are managed by the input management module 536, or other components. In accordance with embodiments of the present invention, the user input can further include moving the screens 104 and 108 relative to one another, between open and closed operating states. Detection of a current operating state and/or a change in operating state can be provided by one or more position sensors 172. In addition, although certain embodiments have been described in connection with operation on a device having first 104 and second 108 screens, the invention is not limited to operation on such a device. For example, a device 100 can have more than two screens, and/or the screen can comprise virtual screens or windows.

[0117] The exemplary systems and methods of this disclosure have been described in relation to the selective presentation of a virtual keyboard 704 in response to a detected operating mode of the device 100 and whether an application having keyboard focus is in view of the user. However, to avoid unnecessarily obscuring the present disclosure, the preceding description omits a number of known structures and devices. This omission is not to be construed as a limitation of the scopes of the claims. Specific details are set forth to provide an understanding of the present disclosure. It should however be appreciated that the present disclosure may be practiced in a variety of ways beyond the specific detail set forth herein.

[0118] Furthermore, while the exemplary aspects, embodiments, and/or configurations illustrated herein show the various components of the system collocated, certain components of the system can be located remotely, at distant portions of a distributed network, such as a LAN and/or the Internet, or within a dedicated system. Thus, it should be appreciated, that the components of the system can be combined in to one or more devices, such as a computer, laptop, netbook, tablet computer, smart phone, mobile device, etc., or collocated on a particular node of a distributed network, such as an analog and/or digital telecommunications network, a packet-switch network, or a circuit-switched network. It will be appreciated from the preceding description, and for reasons of computational efficiency, that the components of the system can be arranged at any location within a distributed network of components without affecting the operation of the system. For example, the various components can be located in a switch such as

a PBX and media server, gateway, in one or more communications devices, at one or more users' premises, or some combination thereof. Similarly, one or more functional portions of the system could be distributed between a telecommunications device(s) and an associated computing device.

[0119] Furthermore, it should be appreciated that the various links connecting the elements can be wired or wireless links, or any combination thereof, or any other known or later developed element(s) that is capable of supplying and/or communicating data to and from the connected elements. These wired or wireless links can also be secure links and may be capable of communicating encrypted information. Transmission media used as links, for example, can be any suitable carrier for electrical signals, including coaxial cables, copper wire and fiber optics, and may take the form of acoustic or light waves, such as those generated during radio-wave and infra-red data communications.

[0120] Also, while the flowcharts have been discussed and illustrated in relation to a particular sequence of events, it should be appreciated that changes, additions, and omissions to this sequence can occur without materially affecting the operation of the disclosed embodiments, configuration, and aspects.

[0121] A number of variations and modifications of the disclosure can be used. It would be possible to provide for some features of the disclosure without providing others.

[0122] For example in one alternative embodiment, while a virtual keyboard 704 is displayed, a first 712 and/or second 716 window can be displayed in a reduced format or size, as compared to when a virtual keyboard 704 is not presented to the user.

[0123] In another alternative embodiment, while a virtual keyboard 704 is displayed, a first 712 and/or second 716 window can be displayed in a reduced format or size, as compared to when a virtual keyboard 704 is not presented to the user.

[0124] In accordance with still other embodiments, the screens 104, 108 may comprise virtual screens or display areas, and selecting different operating modes can include selecting between dual and single virtual screen modes.

[0125] In yet another embodiment, the systems and methods of this disclosure can be implemented in conjunction with a special purpose computer, a programmed microprocessor or microcontroller and peripheral integrated circuit element(s), an ASIC or other integrated circuit, a digital signal processor, a hard-wired electronic or logic circuit such as discrete element circuit,

a programmable logic device or gate array such as PLD, PLA, FPGA, PAL, special purpose computer, any comparable means, or the like. In general, any device(s) or means capable of implementing the methodology illustrated herein can be used to implement the various aspects of this disclosure. Exemplary hardware that can be used for the disclosed embodiments, configurations and aspects includes computers, handheld devices, telephones (e.g., cellular, Internet enabled, digital, analog, hybrids, and others), and other hardware known in the art. Some of these devices include processors (e.g., a single or multiple microprocessors), memory, nonvolatile storage, input devices, and output devices. Furthermore, alternative software implementations including, but not limited to, distributed processing or component/object distributed processing, parallel processing, or virtual machine processing can also be constructed to implement the methods described herein.

[0126] In yet another embodiment, the disclosed methods may be readily implemented in conjunction with software using object or object-oriented software development environments that provide portable source code that can be used on a variety of computer or workstation platforms. Alternatively, the disclosed system may be implemented partially or fully in hardware using standard logic circuits or VLSI design. Whether software or hardware is used to implement the systems in accordance with this disclosure is dependent on the speed and/or efficiency requirements of the system, the particular function, and the particular software or hardware systems or microprocessor or microcomputer systems being utilized.

[0127] In yet another embodiment, the disclosed methods may be partially implemented in software that can be stored on a storage medium, executed on programmed general-purpose computer with the cooperation of a controller and memory, a special purpose computer, a microprocessor, or the like. In these instances, the systems and methods of this disclosure can be implemented as program embedded on personal computer such as an applet, JAVA® or CGI script, as a resource residing on a server or computer workstation, as a routine embedded in a dedicated measurement system, system component, or the like. The system can also be implemented by physically incorporating the system and/or method into a software and/or hardware system.

[0128] Although the present disclosure describes components and functions implemented in the aspects, embodiments, and/or configurations with reference to particular standards and protocols, the aspects, embodiments, and/or configurations are not limited to such standards and

protocols. Other similar standards and protocols not mentioned herein are in existence and are considered to be included in the present disclosure. Moreover, the standards and protocols mentioned herein and other similar standards and protocols not mentioned herein are periodically superseded by faster or more effective equivalents having essentially the same functions. Such replacement standards and protocols having the same functions are considered equivalents included in the present disclosure.

[0129] The present disclosure, in various aspects, embodiments, and/or configurations, includes components, methods, processes, systems and/or apparatus substantially as depicted and described herein, including various aspects, embodiments, configurations embodiments, subcombinations, and/or subsets thereof. Those of skill in the art will understand how to make and use the disclosed aspects, embodiments, and/or configurations after understanding the present disclosure. The present disclosure, in various aspects, embodiments, and/or configurations, includes providing devices and processes in the absence of items not depicted and/or described herein or in various aspects, embodiments, and/or configurations hereof, including in the absence of such items as may have been used in previous devices or processes, e.g., for improving performance, achieving ease and\or reducing cost of implementation.

[0130] The foregoing discussion has been presented for purposes of illustration and description. The foregoing is not intended to limit the disclosure to the form or forms disclosed herein. In the foregoing Detailed Description for example, various features of the disclosure are grouped together in one or more aspects, embodiments, and/or configurations for the purpose of streamlining the disclosure. The features of the aspects, embodiments, and/or configurations of the disclosure may be combined in alternate aspects, embodiments, and/or configurations other than those discussed above. This method of disclosure is not to be interpreted as reflecting an intention that the claims require more features than are expressly recited in each claim. Rather, as the following claims reflect, inventive aspects lie in less than all features of a single foregoing disclosed aspect, embodiment, and/or configuration. Thus, the following claims are hereby incorporated into this Detailed Description, with each claim standing on its own as a separate preferred embodiment of the disclosure.

[0131] Moreover, though the description has included description of one or more aspects, embodiments, and/or configurations and certain variations and modifications, other variations, combinations, and modifications are within the scope of the disclosure, e.g., as may be within the

skill and knowledge of those in the art, after understanding the present disclosure. It is intended to obtain rights which include alternative aspects, embodiments, and/or configurations to the extent permitted, including alternate, interchangeable and/or equivalent structures, functions, ranges or steps to those claimed, whether or not such alternate, interchangeable and/or equivalent structures, functions, ranges or steps are disclosed herein, and without intending to publicly dedicate any patentable subject matter.

What is claimed is:

1. A method, comprising:

providing a device having first and second screens;

in a first operating mode, presenting a virtual keyboard, at least a first portion of a first window, and at least a first portion of a second window, wherein a keyboard focus is on one of the first window or the second window, wherein both the first and second screens are in view of a user;

receiving input ftom the user, wherein the input includes closing the device to place the device in a second operating mode, wherein only one of the first and second screens is in view of the user;

determining whether the user input to place the device in the second operating mode removes the window with keyboard focus ftom the view of the user;

in response to determining that the user input to place the device in the second operating mode does remove the window with keyboard focus from the view of user, presenting at least one of the first portion of the window that does not have keyboard focus and a second portion of the window that does not have keyboard focus, without a presentation of the virtual keyboard.

2. The method of claim 1, wherein in the first operating mode the keyboard focus is on the second window, and wherein in the second operating mode the at least a first portion of the first window is presented without a presentation of the virtual keyboard.

3. The method of claim 1, wherein in the first operating mode the keyboard focus is on the second window, and wherein in the second operating mode the at least a second portion of the first window is presented without a presentation of the virtual keyboard.

4. The method of claim 3, wherein the at least a second portion of the first window presented in the second operating mode includes the first portion of the first window.

5. The method of claim 4, wherein the first and second portions of the first window are presented by the first screen.

6. The method of claim 5, wherein the first screen is a primary screen, and wherein the first screen presents an input window.

7. The method of claim 1, wherein in the first operating mode:

a first portion of the virtual keyboard is presented using a first portion of the first screen,

a second portion of the virtual keyboard is presented using a first portion of the second screen, and wherein the first window is presented using a second portion of the first screen.

8. The method of claim 7, wherein the second window is presented using a second portion of the second screen.

9. The method of claim 1, wherein in the first operating mode the device is in a dual portrait orientation, and wherein in the second operating mode the device is in a single portrait orientation.

10. The method of claim 1, wherein in the first operating mode the device is in a dual landscape orientation, and wherein in the second operating mode the device is in a single landscape orientation.

11. A device, comprising:

a first screen, the first screen including a touch sensitive display area;

a second screen, the second screen including a touch sensitive display area, wherein in a first operating mode both the first screen and the second screen are in view of the user, and wherein in a second operating mode only one of the first and second screens is in view of the user;

memory;

a processor;

application programming stored in the memory and executed by the processor, wherein the application programming is operable to:

present a virtual keyboard and at least a portion of a first window having a keyboard focus;

while presenting the virtual keyboard and at least a first portion of a first window having the keyboard focus, detect a change in a current operating mode;

in response to detecting the change in the current operating mode, determining whether the detected change in a current operating mode has removed the at least a portion of the first window having the keyboard focus from the view of the user;

in response to determining that the change in a current operating mode has removed the at least a portion of the first window having the keyboard focus from the view of the user, presenting at least one of the first portion and a second portion of the first window without presenting the virtual keyboard.

12. The device of claim 11, further comprising:

a hinge, wherein the hinge interconnects the first and second screens, wherein in the first operating mode the device is in an open state, and wherein in the second operating mode the device is in a closed state.

13. The device of claim 12, further comprising:

a position sensor, wherein the position sensor is operable to provide a signal indicating whether the device is in the open state or the closed state.

14. The device of claim 13, wherein in the first operating mode the virtual keyboard is presented by a first portion of the first screen and a first portion of the second screen, and wherein the at least a first portion of the window is presented by at least one of a second portion of the first screen and a second portion of the second screen.

15. The device of claim 14, wherein in the first operating mode the at least a first portion of the first window is presented by the second portion of the first screen and the at least a first portion of the second window is presented by the second portion of the first screen, and wherein in the second operating mode a second portion of the first window, the second portion including the first portion of the first window, is presented by the first screen, and no virtual keyboard is presented.

16. A computer readable medium having stored thereon computer-executable instructions, the computer executable instructions causing a processor to execute a method for providing a virtual keyboard, the computer-executable instructions comprising:

instructions to determine an operating mode of a device having at least a first touch sensitive screen and a second touch sensitive screen;

instructions to, in a first operating mode, display a virtual keyboard, at least a portion of a first window having a keyboard focus, and at least a portion of a second window;

instructions to detect a change from the first operating mode to a second operating mode, and in response to the detected change in operating mode:

determine whether the change from the first operating mode to the second operating mode has removed the at least a first portion of the first window from a view of the user;

in response to determining that the at least a first portion of the first window has been removed from the view of the user, present at least one of the first portion and a second portion of the second window, without a presentation of the virtual keyboard.

17. The computer readable medium of claim 17, wherein in the first operating mode both of the first and second touch sensitive screens are operative, and wherein in the second operating mode only one of the first and second touch sensitive screens is operative.

18. The computer readable medium of claim 17, wherein the change from the first operating mode to the second operating mode includes the user closing the device.

19. The computer readable medium of claim 18, wherein the change from the first operating mode to the second operating mode is detected in response to a signal from a position sensor received by the processor.

20. The computer readable medium of claim 17, wherein in the first operating mode the virtual keyboard is presented using a first portion of the first touch sensitive screen and a first portion of the second touch sensitive screen, wherein the first portion of the first window is presented by a second portion of a first one of the first and second touch sensitive screens, and

wherein at least a first portion of a second window is presented by a second portion of the second touch sensitive screen.

Attorney Docket No. 6583-62

ABSTRACT

Methods and devices for selectively presenting a virtual keyboard are provided. More particularly, a change in the operating mode of a multiple screen device between a multiple screen operating mode and a single screen operating mode can be determined. Moreover, a change in the operating mode can effect a change in a presentation of a virtual keyboard. More particularly, a presentation of a virtual keyboard can be discontinued in response to a change in the operating mode of the device, where an application that had keyboard focus prior to the change is, after the change, no longer in view of the user.

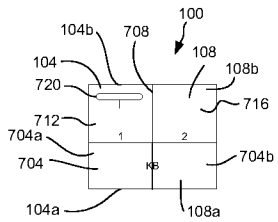


Fig. 7A

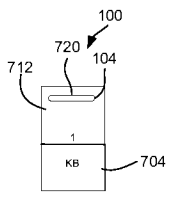


Fig. 7B

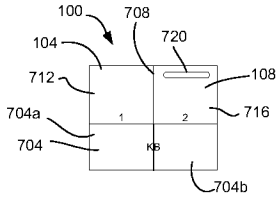


Fig. 8A

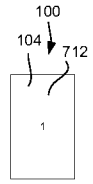


Fig. 8B

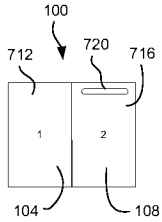


Fig. 8C

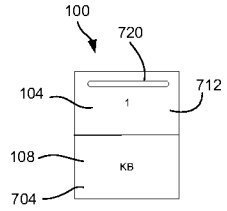


Fig. 9A

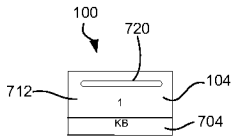


Fig. 9B

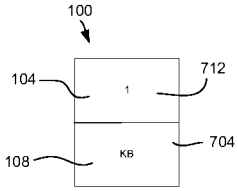


Fig. 10A

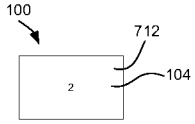


Fig. 10B

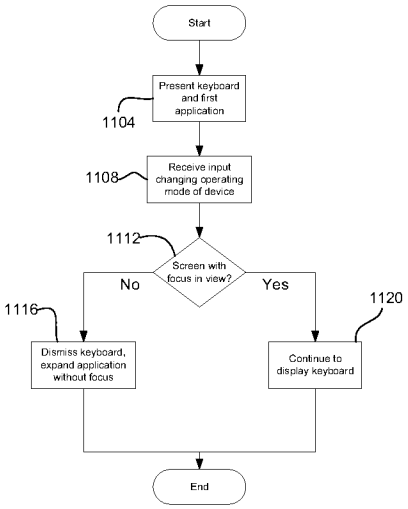


Fig. 11

PINCH GESTURE TO SWAP WINDOWS

[01] The device 100 manages desktops and/or windows with at least one window stack 1700, 1728, as shown in FIGS. 14A and 14B. A window stack 1700, 1728 is a logical arrangement of active and inactive windows for a dual screen device. For example, the window stack 1700, 1728 may be logically similar to a deck of cards, where one or more windows or desktops are arranged in order, as shown in FIGS. 14A and 14B. An active window is a window that is currently being displayed on at least one of the two touch sensitive displays 110, 114. For example, windows 104 and 108 are active windows and are displayed on touch sensitive displays 110 and 114. An inactive window is a window that was opened and displayed but is now "behind" an active window and not being displayed. In embodiments, an inactive window may be for an application that is suspended, and thus, the window is not displaying active content. For example, windows 1712, 1716, 1720, and 1724 are inactive windows.

[02] A window stack 1700, 1728 may have various arrangements or organizational structures. In the embodiment shown in Fig. 14A, the device 100 includes a first stack 1760 associated with a first touch sensitive display 110 and a second stack associated with a second touch sensitive display 114. Thus, each touch sensitive display 110, 114 can have an associated window stack 1760, 1764. These two window stacks 1760, 1764 may have different numbers of windows arranged in the respective stacks 1760, 1764. Further, the two window stacks 1760, 1764 can also be identified differently and managed separately. Thus, the first window stack 1760 can be arranged in order from a first window 1704 to a next window 1720 to a last window 1724. Likewise, the second stack 1764 can be arranged from a first window 1708 to a next window 1712 to a last window 1716. A logical data structure for managing the two window stacks 1760, 1764 may be as described in conjunction with Fig. 14.

[03] Another arrangement for a window stack 1728 is shown in Fig. 14B. In this embodiment, there is a single window stack 1728 for both touch sensitive displays 110, 114. Thus, the window stack 1728 is arranged from a first window 1744 to a last window 1756. A window can be arranged in a position among all windows without an association to a specific touch sensitive display 110, 114. In this embodiment, a window is in the order of windows. Further, at least one window is identified as being active. For example, a single window may be rendered in two portions 1732 and 1736 that are displayed on the first touch sensitive screen 110 and the second touch sensitive screen 114. The single window may only occupy a single position in the window stack 1728 although it is displayed on both displays 110, 114.

[04] When a new window is opened, the newly activated window is positioned within the stack. Where and how the window is positioned within the stack can be a function of the orientation of the device 100, the context of what programs, functions, software, etc. are being executed on the device 100, how the stack is positioned when the new window is opened, etc. To insert the window in the stack, the position in the stack for the window is determined and the touch sensitive display 110, 114 to which the window is associated may also be determined. With this information, a logical data structure for the window can be created and stored. When user interface or other events or activities change the arrangement of windows, the window stack(s) can be changed to reflect the change in arrangement. It should be noted that these same concepts described above can be used to manage the one or more desktops for the device 100.

[05] FIG. 16 depicts a further window stacking configuration. A plurality of windows 1, 2, 3, 4, 5, 6, 7, and 8, whether from the same or different multi-screen or single-screen applications, are depicted. The touch sensitive display 110 currently has window 4 in the active display position while touch sensitive display 114 currently has window 5 in the active display position. The stack for touch sensitive display 110, from top to bottom, has window 4 in the active display position and windows 3, 2, and 1 positioned behind it. The stack for touch sensitive display 114, from top to bottom, has window 5 in the active display position and windows 6, 7, and 8 positioned behind it.

[06] The desktops D1, D2, D3, D4, D5, and D6 are positioned behind the window stacks. The desktops can be viewed as a desktop stack different from the window stack. Viewed in this manner, the touch sensitive display 110 has a corresponding desktop stack comprising desktops D3, D2, and D1,with desktop D1 being at the bottom 2300 stack position and with desktop D3 at the top stack position being capable of being displayed with window 4 (depending on the window position and size (whether maximized or minimized), and the touch sensitive display 114 has a corresponding desktop stack has a corresponding desktop stack comprising desktops D4, D5, and D6,with desktop D6 being at the bottom 2304 stack position and with desktop D4 at the top stack position being capable of being displayed with window 5 (depending on the window position and size (whether maximized or minimized). Conceptually, the desktops can be viewed as a canvas divided, in this example, into six sections of which two can be displayed,

at any one time, the touch sensitive displays 110, 114. When the device 100 is in the closed state, this conceptual model, in one configuration, persists. In this configuration, only one window and desktop stacks can be seen (that corresponding to the primary screen) but the other window and desktop stacks are virtual; that is, they are maintained in memory but cannot be seen because the secondary screen is not enabled.

[07] A logical data structure 1800 for managing the arrangement of windows or desktops in a window stack is shown in FIG. 17. The logical data structure 1800 can be any data structure used to store data whether an object, record, file, etc. The logical data structure 1800 can be stored in any type of database or data storage system, regardless of protocol or standard. In embodiments, the logical data structure 1800 includes one or more portions, fields, attributes, etc. that store data in a logical arrangement that allows for easy storage and retrieval of the information. Hereinafter, these one or more portions, fields, attributes, etc. shall be described simply as fields. The fields can store data for a window identifier 1804, a stack identifier 1808, a stack position identifier 1812, a display identifier 1816, and/or an active indicator 1820. Each window in a window stack can have an associated logical data structure 1800. While only a single logical data structure 1800 is shown in Fig. 17, there may be more or fewer logical data structures 1800 used with a window stack (based on the number of windows or desktops in the stack), as represented by ellipses 1824. Further, there may be more or fewer fields than those shown in Fig. 17, as represented by ellipses 1828.

[08] A window identifier 1804 can include any identifier (ID) that uniquely identifies the associated window in relation to other windows in the window stack. The window identifier 1804 can be a globally unique identifier (GUID), a numeric ID, an alphanumeric ID, or other type of identifier. In embodiments, the window identifier 1804 can be one, two, or any number of digits based on the number of windows that can be opened. In alternative embodiments, the size of the window identifier 1804 may change based on the number of windows opened. While the window is open, the window identifier 1804 may be static and remain unchanged.

[09] A stack identifier 1808 can include an identifier to identify which of a first stack and a second stack the window is associated. For example, in the arrangement 1700 shown in Fig. 14A, a window can be associated with one of two stacks 1760 or 1764. The stack identifier 1808 identifies which stack of the two stacks the window is associated. The stack identifier 1808 can be a GUID, a numeric ID, an alphanumeric ID, or other type of identifier. However, the stack

identifier 1808, in embodiments, is either a one or a two representing one of the two stacks. The stack identifier 1808 can be dynamic as it may change if the window or desktop is moved from one display to another display and, thus, changes stacks.

[010] A stack position identifier 1812 can be any identifier that can identify the position in the stack for the window. The stack position identifier 1812 can be a GUID, a numeric ID, an alphanumeric ID, or other type of identifier. Each window or desktop can include a stack position identifier 1812. For example, as shown in Fig. 14A, window 1 1704 in stack 1 1760 can have a stack position identifier 1812 of 1 identifying that window 1704 is the first window in the stack 1760 and the active window. Similarly, window 6 1724 can have a stack position identifier 1812 of 3 representing that window 1724 is the third window in the stack 1760. Window 2 1708 can also have a stack position identifier 1812 of 1 representing that window 1708 is the first window in the second stack 1764. As shown in Fig. 14B, window 1 1744 can have a stack position identifier 1812 of 1, window 3, rendered in portions 1732 and 1736, can have a stack position identifier 1812 of 3, and window 6 1756 can have a stack position identifier 1812 of 6. Thus, depending on the type of stack, the stack position identifier 1812 can represent a window's location in the stack.

[011] A display identifier 1816 can identify that the window or desktop is associated with either the first display 110 or the second display 114. While this display identifier 1816 may not be needed for a dual stack system, as shown in Fig. 14A, the display identifier 1816 can indicate whether a window in the serial stack of Fig. 14B is displayed on a particular display. Thus, window 3 may have two portions 1732 and 1736 in Fig. 14B. The first portion 1732 may have a display indicator 1816 for the first display while the second portion 1736 may have a display indicator 1816 for the second display 114. However, in alternative embodiments, the window may have two display indicators 1816 that represent that the window is displayed on both of the displays 110, 114.

[012] Similar to the display identifier 1816, an active indicator 1820 may not be needed with the dual stack system of Fig. 14A, as the window in stack position 1 is active and displayed. In the system of Fig. 14B, the active indicator 1820 can indicate which window(s) in the stack is being displayed. Thus, window 3 may have two portions 1732 and 1736 in Fig. 17. The first portion 1732 may have an active indicator 1820 while the second portion 1736 may also have an active indicator 1820. However, in alternative embodiments, window 3 may have a single active

indicator 1820. The active indicator 1820 can be a simple flag or bit that represents that the window is active or displayed.

[013] Various window navigation configurations will now be discussed with reference to FIGS. 7-13. Generally, the window navigation logic enables a first window to be moved from a (first or origin or source) touch sensitive display 110 to a (second or target) touch sensitive display 114, and a (minimized) second window to be moved from the (second or origin or source) touch sensitive display 114 to the (first or target) touch sensitive display 110. The windows are considered to be minimized in that they occupy only one of the primary and secondary screens 104 and 108. In one configuration, a "pinch" gesture (such as shown in FIG. 4D) (or some other defined gesture) initiates swapping the respective positions of first and second windows in active display positions on the adjacent touch sensitive displays 110 and 114. Only the windows in the active display positions are affected; the remaining members of the respective stacks of each touch sensitive display remain unchanged. In other words, the stack positions of the remaining members of the window stacks for each touch sensitive display are the same after the swap as before the swap. The swap operation can be performed in both portrait and landscape display orientations. In one configuration, swapping swaps first and second windows of a common multi-screen application between touch sensitive displays 110 and 114. In one configuration, swapping can be used to swap a window of a single-screen application from one touch sensitive display to the other. This is true even when only one window is displayed on the touch sensitive displays 110 and 114 or when one of the touch-sensitive displays 110 and 114 displays a desktop and the other displays the window. In one configuration, swapping can be used to swap first and second windows from first and second applications, respectively, between touch sensitive displays 110 and 114.

[014] Referring to FIG. 7A, a pinch gesture is received by the gesture capture regions 120 and 124 or by the touch sensitive displays 110 and 114 while the device is in the portrait display orientation. The two contacts 700A and 700B of the pinch gesture occur at substantially the same time or are temporally overlapping. The touch sensitive displays 110 and 114 are currently displaying first and second desktops D1 and D2. In one configuration, the pinch gesture is not effective to change the respective positions of the desktops (FIG. 7B). In another configuration, the pinch gesture causes the first desktop D1 to be displayed by the touch sensitive display 114,

and the second desktop D2 to be displayed by the touch sensitive display 110 (not shown). Due to the user desktop confusion that can be caused by this configuration, it is not preferred.

[015] Referring to FIG. 8A, a pinch gesture is received by the gesture capture regions 120 and 124 or by the touch sensitive displays 110 and 114. The two contacts 700A and 700B of the pinch gesture occur at substantially the same time or are temporally overlapping. The touch sensitive displays 110 and 114 are currently displaying a first window 1 and second desktop D2, respectively. The first window 1 is in the active display position ahead of the first desktop D1. As a result of the pinch gesture, the display position of the first window 1 has moved from the touch sensitive display 110 to touch sensitive display 114, causing the first desktop to advance to the active display position for the touch sensitive display 110, and the second desktop D2 to be moved to an inactive display position behind the first window 1 (FIG. 8B).

[016] Referring to FIG. 9A, a further pinch gesture is received by the gesture capture regions 120 and 124 or by the touch sensitive displays 110 and 114. The touch sensitive displays 110 and 114 are currently displaying a first desktop D1 and first window 1, respectively. The first window 1 is in the active display position ahead of the second desktop D2. As a result of the pinch gesture, the display position of the first window 1 has moved from the touch sensitive display 114 to touch sensitive display 110, causing the second desktop to advance to the active display position for the touch sensitive display 114, and the first desktop D1 to be moved to an inactive display position behind the first window 1 (FIG. 9B).

[017] Referring to FIG. 10A, a pinch gesture is received by the gesture capture regions 120 and 124 or by the touch sensitive displays 110 and 114. The touch sensitive displays 110 and 114 are currently displaying a first window 1 and second window 2, respectively. In one configuration, the first and second windows 1, 2 are controlled by different single-screen applications. In other configurations, the first and second windows are controlled by a common multi-screen application or by different multi-screen and single-screen applications. The first window 1 is in the active display position ahead of the first desktop D1. The second window 2 is in the active display position ahead of the second desktop D2. As a result of the pinch gesture, the display position of the first window 1 has moved from the touch sensitive display 110 to touch sensitive display 114, and the display position of the second window 2 has moved from the touch sensitive display 114 to touch sensitive display 110 (FIG. 10B).

[018] Referring to FIG. 11A, a further pinch gesture is received by the gesture capture regions 120 and 124 or by the touch sensitive displays 110 and 114. The touch sensitive displays 110 and 114 are currently displaying the second window 2 and first window 1, respectively. The second window is now in the active display position ahead of the first desktop D1, and the first window 1 is in the active display position ahead of the second desktop D2. As a result of the pinch gesture, the display position of the first window 1 has moved from the touch sensitive display 114 to touch sensitive display 110, and the display position of the second window 2 has moved from the touch sensitive display 110 to touch sensitive display 114.

[019] Referring to FIG. 12A, the device 100 is now rotated to a landscape display orientation. The touch sensitive display 110 is currently displaying the minimized first window 1 ahead of the first desktop D1, and the touch sensitive display 114 is currently displaying the minimized second window 2 ahead of the second desktop D2. In response to the gesture, the display position of the first window 1 has moved from the touch sensitive display 110 to touch sensitive display 114 (ahead of the second desktop D2), and the display position of the second window 2 has moved from the touch sensitive display 114 to touch sensitive display 110 (ahead of the first desktop D1) (FIG. 12B).

[020] Referring to FIG. 13, middleware 520, particularly one or more of the Multi-Display Management (MDM) class 524, a Surface Cache class 528, a Window Management class 532, an Activity Management class 536, and an Application Management class 540, independently or collectively, detect receipt of a user gestures (step 1300 of FIG. 13), in response, swap displayed images between the first and second touch sensitive displays (or move a displayed image from one touch sensitive display to the other), and the logic terminates in step 1306.

[021] The exemplary systems and methods of this disclosure have been described in relation to communication devices. However, to avoid unnecessarily obscuring the present disclosure, the preceding description omits a number of known structures and devices. This omission is not to be construed as a limitation of the scopes of the claims. Specific details are set forth to provide an understanding of the present disclosure. It should however be appreciated that the present disclosure may be practiced in a variety of ways beyond the specific detail set forth herein.

[022] Furthermore, while the exemplary aspects, embodiments, and/or configurations illustrated herein show the various components of the system collocated, certain components of the system can be located remotely, at distant portions of a distributed network, such as a LAN

and/or the Internet, or within a dedicated system. Thus, it should be appreciated, that the components of the system can be combined in to one or more devices, such as a communication device, or collocated on a particular node of a distributed network, such as an analog and/or digital telecommunications network, a packet-switch network, or a circuit-switched network. It will be appreciated from the preceding description, and for reasons of computational efficiency, that the components of the system can be arranged at any location within a distributed network of components without affecting the operation of the system. For example, the various components can be located in a switch such as a PBX and media server, gateway, in one or more communications devices, at one or more users' premises, or some combination thereof. Similarly, one or more functional portions of the system could be distributed between a telecommunications device(s) and an associated computing device.

[023] Furthermore, it should be appreciated that the various links connecting the elements can be wired or wireless links, or any combination thereof, or any other known or later developed element(s) that is capable of supplying and/or communicating data to and from the connected elements. These wired or wireless links can also be secure links and may be capable of communicating encrypted information. Transmission media used as links, for example, can be any suitable carrier for electrical signals, including coaxial cables, copper wire and fiber optics, and may take the form of acoustic or light waves, such as those generated during radio-wave and infra-red data communications.

[024] Also, while the flowcharts have been discussed and illustrated in relation to a particular sequence of events, it should be appreciated that changes, additions, and omissions to this sequence can occur without materially affecting the operation of the disclosed embodiments, configuration, and aspects.

[025] A number of variations and modifications of the disclosure can be used. It would be possible to provide for some features of the disclosure without providing others.

[026] For example in one alternative embodiment, gestures other than a pinch gesture are used to effect the swap function.

[027] In yet another embodiment, the systems and methods of this disclosure can be implemented in conjunction with a special purpose computer, a programmed microprocessor or microcontroller and peripheral integrated circuit element(s), an ASIC or other integrated circuit, a digital signal processor, a hard-wired electronic or logic circuit such as discrete element circuit,

a programmable logic device or gate array such as PLD, PLA, FPGA, PAL, special purpose computer, any comparable means, or the like. In general, any device(s) or means capable of implementing the methodology illustrated herein can be used to implement the various aspects of this disclosure. Exemplary hardware that can be used for the disclosed embodiments, configurations and aspects includes computers, handheld devices, telephones (e.g., cellular, Internet enabled, digital, analog, hybrids, and others), and other hardware known in the art. Some of these devices include processors (e.g., a single or multiple microprocessors), memory, nonvolatile storage, input devices, and output devices. Furthermore, alternative software implementations including, but not limited to, distributed processing or component/object distributed processing, parallel processing, or virtual machine processing can also be constructed to implement the methods described herein.

[028] In yet another embodiment, the disclosed methods may be readily implemented in conjunction with software using object or object-oriented software development environments that provide portable source code that can be used on a variety of computer or workstation platforms. Alternatively, the disclosed system may be implemented partially or fully in hardware using standard logic circuits or VLSI design. Whether software or hardware is used to implement the systems in accordance with this disclosure is dependent on the speed and/or efficiency requirements of the system, the particular function, and the particular software or hardware systems or microprocessor or microcomputer systems being utilized.

[029] In yet another embodiment, the disclosed methods may be partially implemented in software that can be stored on a storage medium, executed on programmed general-purpose computer with the cooperation of a controller and memory, a special purpose computer, a microprocessor, or the like. In these instances, the systems and methods of this disclosure can be implemented as program embedded on personal computer such as an applet, JAVA® or CGI script, as a resource residing on a server or computer workstation, as a routine embedded in a dedicated measurement system, system component, or the like. The system can also be implemented by physically incorporating the system and/or method into a software and/or hardware system.

[030] Although the present disclosure describes components and functions implemented in the aspects, embodiments, and/or configurations with reference to particular standards and protocols, the aspects, embodiments, and/or configurations are not limited to such standards and protocols.

Other similar standards and protocols not mentioned herein are in existence and are considered to be included in the present disclosure. Moreover, the standards and protocols mentioned herein and other similar standards and protocols not mentioned herein are periodically superseded by faster or more effective equivalents having essentially the same functions. Such replacement standards and protocols having the same functions are considered equivalents included in the present disclosure.

[031] The present disclosure, in various aspects, embodiments, and/or configurations, includes components, methods, processes, systems and/or apparatus substantially as depicted and described herein, including various aspects, embodiments, configurations embodiments, subcombinations, and/or subsets thereof. Those of skill in the art will understand how to make and use the disclosed aspects, embodiments, and/or configurations after understanding the present disclosure. The present disclosure, in various aspects, embodiments, and/or configurations, includes providing devices and processes in the absence of items not depicted and/or described herein or in various aspects, embodiments, and/or configurations hereof, including in the absence of such items as may have been used in previous devices or processes, e.g., for improving performance, achieving ease and\or reducing cost of implementation.

[032] The foregoing discussion has been presented for purposes of illustration and description. The foregoing is not intended to limit the disclosure to the form or forms disclosed herein. In the foregoing Detailed Description for example, various features of the disclosure are grouped together in one or more aspects, embodiments, and/or configurations for the purpose of streamlining the disclosure. The features of the aspects, embodiments, and/or configurations of the disclosure may be combined in alternate aspects, embodiments, and/or configurations other than those discussed above. This method of disclosure is not to be interpreted as reflecting an intention that the claims require more features than are expressly recited in each claim. Rather, as the following claims reflect, inventive aspects lie in less than all features of a single foregoing disclosed aspect, embodiment, and/or configuration. Thus, the following claims are hereby incorporated into this Detailed Description, with each claim standing on its own as a separate preferred embodiment of the disclosure.

[033] Moreover, though the description has included description of one or more aspects, embodiments, and/or configurations and certain variations and modifications, other variations, combinations, and modifications are within the scope of the disclosure, e.g., as may be within the

skill and knowledge of those in the art, after understanding the present disclosure. It is intended to obtain rights which include alternative aspects, embodiments, and/or configurations to the extent permitted, including alternate, interchangeable and/or equivalent structures, functions, ranges or steps to those claimed, whether or not such alternate, interchangeable and/or equivalent structures, functions, ranges or steps are disclosed herein, and without intending to publicly dedicate any patentable subject matter.

What is claimed is:

1. A method, comprising:

receiving, by at least one of a gesture capture region and a touch sensitive display, a gesture while a first touch sensitive display of a communication device is displaying a first displayed image and a second touch sensitive display of the communication device is displaying a second displayed image; and

in response, ceasing, by a microprocessor, to display the first displayed image on the first touch sensitive display and commencing to display the first displayed image on the second touch sensitive display and ceasing to display a second displayed image on the second touch sensitive display and commencing to display the second displayed image on the first touch sensitive display, wherein the first displayed image is at least one of a desktop and window of an application.

2. The method of claim 1, wherein the gesture is a pinch gesture and wherein the first touch sensitive display ceases to display the first displayed image at substantially the same time as the second touch display ceases to display the second displayed image.

3. The method of claim 1, wherein the first displayed image is a window, wherein the gesture is received by the gesture capture region, and wherein the gesture capture region is unable to display any displayed image.

4. The method of claim 1, wherein the first touch sensitive display starts to display the second displayed image at substantially the same time as the first touch display starts to display the second displayed image.

5. The method of claim 1, wherein each of the first and second touch sensitive displays has a respective stack of displayed images, with the first and second displayed images being in active display positions and wherein, with the exception of the first and second displayed images, the respective stacks of displayed images of the first and second touch sensitive displays remain unchanged.

6. The method of claim 1, wherein the first and second displayed images are first and second windows, respectively, and wherein the first window is controlled by a first single-screen application and the second window is controlled by a second single-screen application.

7. The method of claim 1, wherein the first and second displayed images are first and second windows, respectively, and wherein the first and second windows are controlled by a common multi-screen application.

8. A non-transient computer readable medium comprising processor executable instructions operable to perform the steps of claim 1.

9. A method, comprising:

displaying, by a first touch sensitive display, a window and, by a second touch sensitive display, a second desktop;

receiving, by at least one of a gesture capture region and a touch sensitive display, a gesture; and

in response, moving, by a microprocessor, the window from the first touch sensitive display to a second touch sensitive display, whereby the first touch sensitive display displays a first desktop and the second touch sensitive display displays the window.

10. The method of claim 9, wherein the window is controlled by a single-screen application and wherein the gesture is a pinch gesture.

11. The method of claim 9, wherein each of the first and second touch sensitive displays has a respective stack of images, with the window is in an active display position, and wherein, with the exception of the window, the respective stacks of images of the first and second touch sensitive displays remain unchanged.

12. A non-transient computer readable medium comprising processor executable instructions operable to perform the steps of claim 9.

13. A dual display communication device operable to perform the steps of claim 9.

14. A dual display communication device, comprising:

a gesture capture region to receive a gesture;

a first touch sensitive display to receive a gesture and display displayed images, wherein the displayed images are at least one of a desktop and window of an application;

a second touch sensitive display to receive a gesture and display displayed images; and

middleware operable to perform at least one of the following operations:

receiving, by at least one of a gesture capture region and a touch sensitive display, a gesture while a first touch sensitive display of a communication device is displaying a first displayed image and a second touch sensitive display of the communication device is displaying a second displayed image; and

in response, ceasing to display the first displayed image on the first touch sensitive display and commencing to display the first displayed image on the second touch sensitive display and ceasing to display a second displayed image on the second touch sensitive display and commencing to display the second displayed image on the first touch sensitive display, wherein the first displayed image is at least one of a desktop and window of an application.

15. The device of claim 14, wherein the gesture is a pinch gesture and wherein the first touch sensitive display ceases to display the first displayed image at substantially the same time as the second touch display ceases to display the second displayed image.

16. The device of claim 14, wherein the first displayed image is a window, wherein the gesture is received by the gesture capture region, and wherein the gesture capture region is unable to display any displayed image.

17. The device of claim 14, wherein the first touch sensitive display starts to display the second displayed image at substantially the same time as the first touch display starts to display the second displayed image.

18. The device of claim 14, wherein each of the first and second touch sensitive displays has a respective stack of displayed images, with the first and second displayed images being in active display positions and wherein, with the exception of the first and second displayed images, the respective stacks of displayed images of the first and second touch sensitive displays remain unchanged.

19. The device of claim 14, wherein the first and second displayed images are first and second windows, respectively, and wherein the first window is controlled by a first single-screen application and the second window is controlled by a second single-screen application.

20. The device of claim 14, wherein the first and second displayed images are first and second windows, respectively, and wherein the first and second windows are controlled by a common multi-screen application.

ABSTRACT

[034] The disclosed method and device are directed to a communication device that receives, by a gesture capture region and/or a touch sensitive display, a gesture while a first touch sensitive display is displaying a first displayed image and a second touch sensitive display is displaying a second displayed image and, in response, ceasing to display the first displayed image on the first touch sensitive display and commencing to display the first displayed image on the second touch sensitive display and ceasing to display a second displayed image on the second touch sensitive display and commencing to display the second displayed image on the first touch sensitive display.

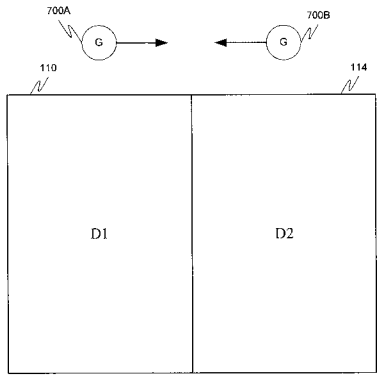


FIG. 7A

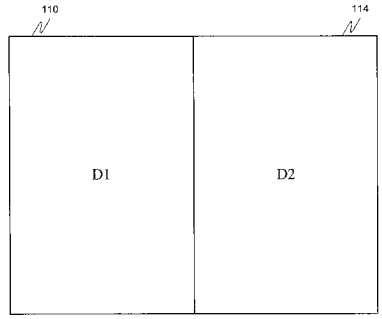


FIG. 7B

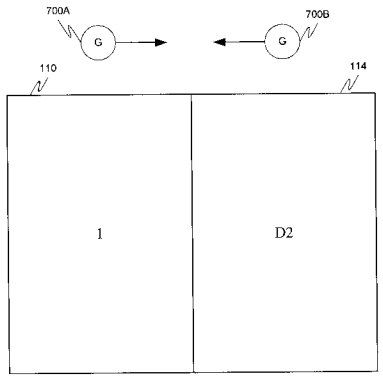


FIG. 8A

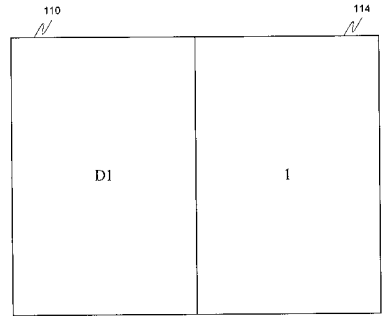


FIG. 8B

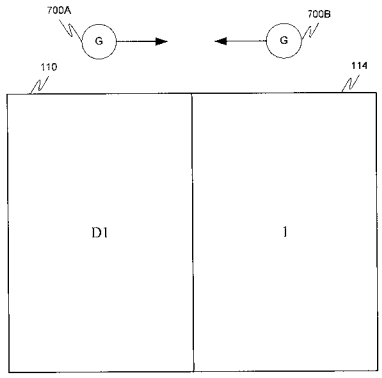


FIG. 9A

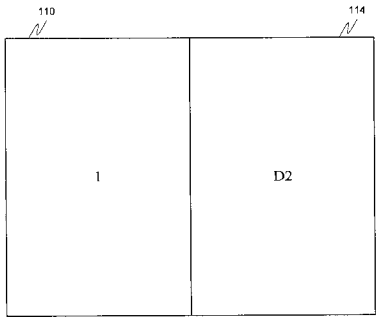


FIG. 9B

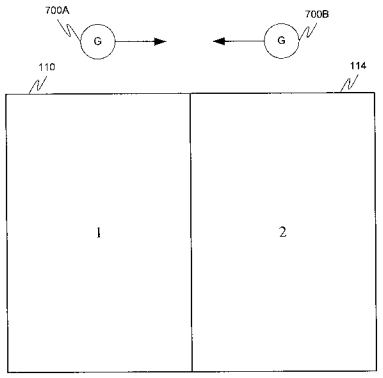


FIG. 10A

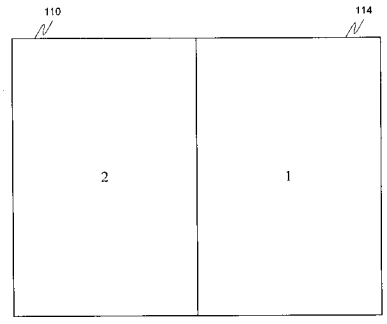


FIG. 10B

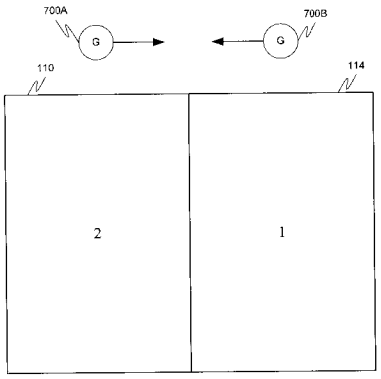


FIG. 11A

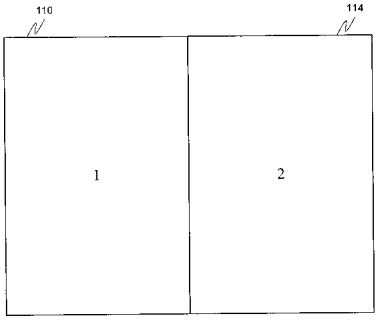


FIG. 11B

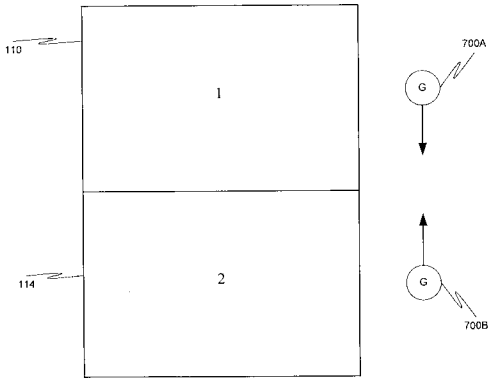


FIG. 12A

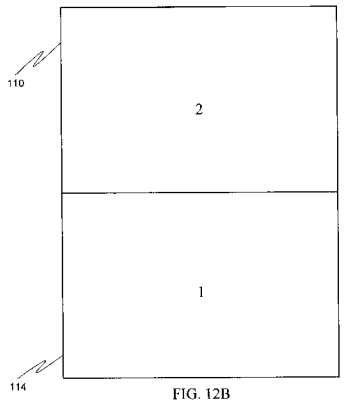


FIG. 12B

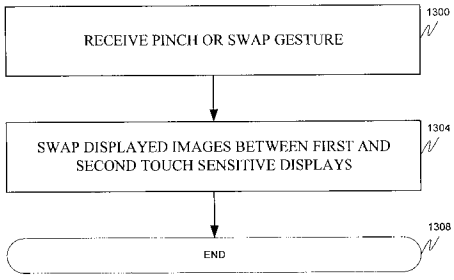


FIG. 13

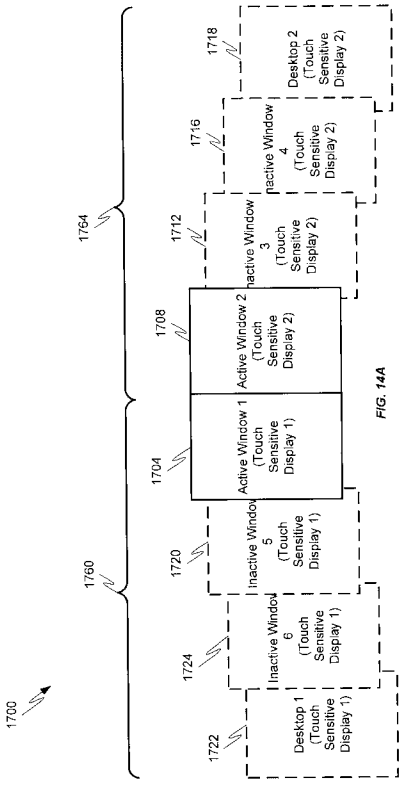


FIG. 14A

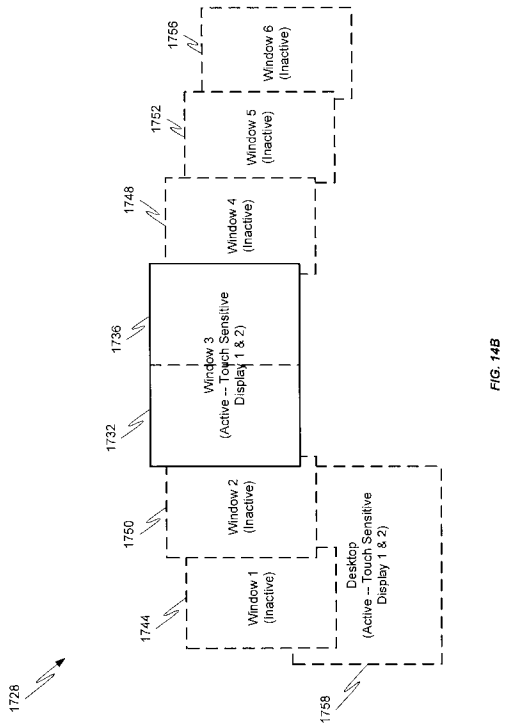


FIG. 14B

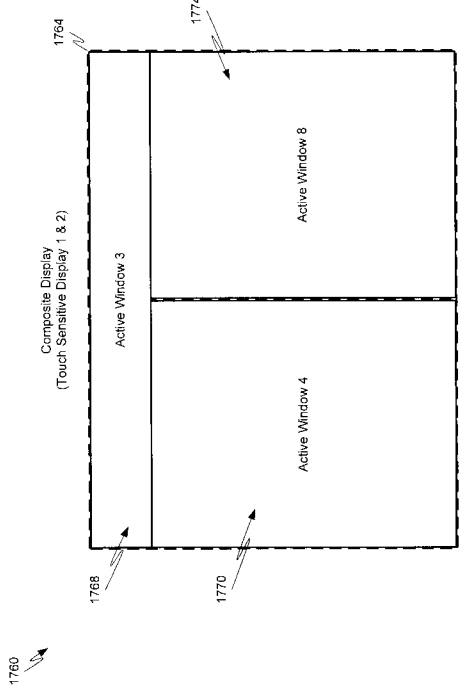


FIG. 14C

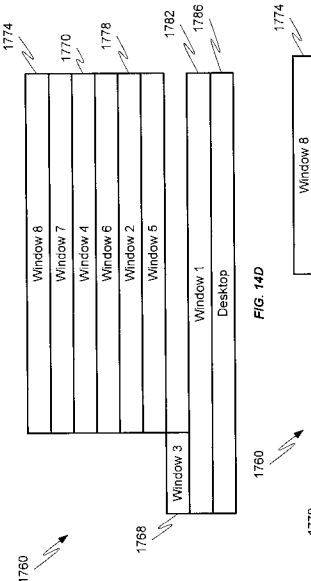


FIG. 14D

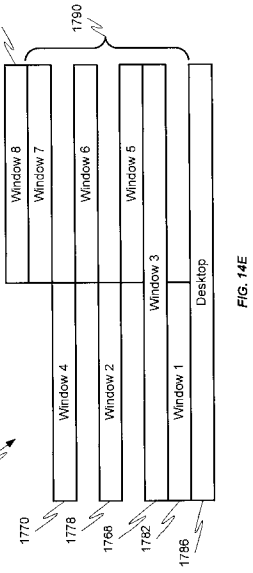


FIG. 14E

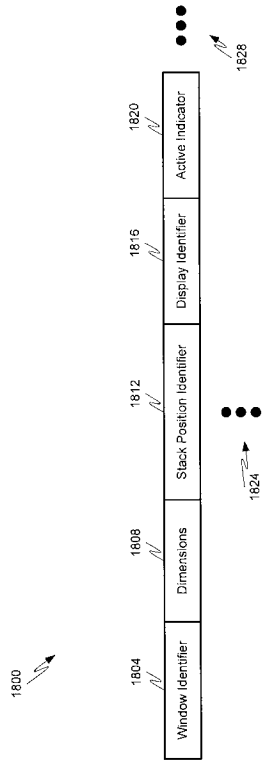


FIG. 15

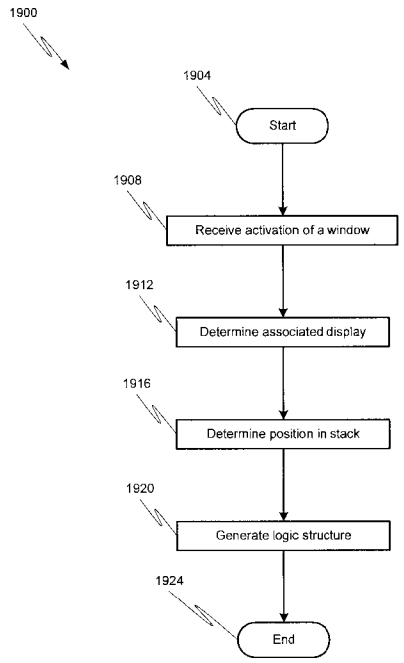

1900
1904
Start
1908
Receive activation of a window
1912
Determine associated display
1916
Determine position in stack
1920
Generate logic structure
1924
End

FIG. 16

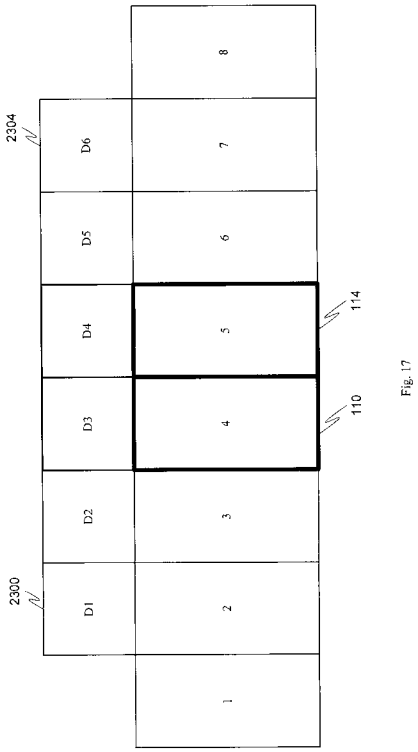

2304
D6
D5
D4
D3
D2
D1
2300
8
7
6
5
4
3
2
1
114
110

Fig. 17

FLICK MOVE GESTURE IN USER INTERFACE

[0100] An alternative output configuration may be accommodated by the device 100 being in a second state. Specifically, FIG. 6C depicts a third portrait configuration where data is displayed simultaneously on both the primary screen 104 and the secondary screen 108. The third portrait configuration may be referred to as a Dual-Portrait (PD) output configuration. In the PD output configuration, the touch sensitive display 110 of the primary screen 104 depicts data in the first portrait configuration 604 while the touch sensitive display 114 of the secondary screen 108 depicts data in the second portrait configuration 608. The simultaneous presentation of the first portrait configuration 604 and the second portrait configuration 608 may occur when the device 100 is in an open portrait state 320. It should be appreciated that it may also be possible to transition the device 100 from the simultaneous display of configurations 604, 608 to any other configuration described herein depending upon which state the device 100 is moved.

[0101] FIGS. 6D and 6E depicts two further output configurations of the device 100 being in a third state. Specifically, FIG. 6D depicts the device 100 being in a closed landscape state 340 where the data is displayed on the primary screen 104. In this example, the device 100 displays data via the touch sensitive display 110 in a first landscape configuration 612. Much like the other configurations described herein, the first landscape configuration 612 may display a desktop, a home screen, one or more windows displaying application data, or the like.

[0102] FIG. 6E depicts the device 100 still being in the closed landscape state 340, but instead data is displayed on the secondary screen 108. In this example, the device 100 displays data via the touch sensitive display 114 in a second landscape configuration 616. It may be possible to display similar or different data in either the first or second portrait configuration 612, 616. It may also be possible to transition between the first landscape configuration 612 and second landscape configuration 616 by providing the device 100 with one or both of a twist and tap gesture or a flip and slide gesture. **O**ther suitable gestures may also be employed to transition between configurations. Furthermore, it may also be possible to transition the device 100 from the first or second landscape configuration 612, 616 to any other configuration described herein depending upon which state the device 100 is moved.

[0103] FIG. 6F depicts a third landscape configuration where data is displayed simultaneously on both the primary screen 104 and the secondary screen 108. The third landscape configuration may be referred to as a Dual-Landscape (LD) output configuration. In the LD output configuration, the touch sensitive display 110 of the primary screen 104 depicts data in the first landscape configuration 612 while the touch sensitive display 114 of the secondary screen 108 depicts data in the second landscape configuration 616. The simultaneous presentation of the first landscape configuration 612 and the second landscape configuration 616 may occur when the device 100 is in an open landscape state 340. It should be appreciated that it may also be possible to transition the device 100 from the simultaneous display of configurations 612, 616 to any other configuration described herein depending upon which state the device 100 is moved.

[0104] FIGS. 6G and 6H depict two views of a device 100 being in yet another state. Specifically, the device 100 is depicted as being in an easel state 312. FIG. 6G shows that a first easel output configuration 618 may be displayed on the touch sensitive display 110. FIG. 6H shows that a second easel output configuration 620 may be displayed on the touch sensitive display 114. The device 100 may be configured to depict either the first easel output configuration 618 or the second easel output configuration 620 individually. Alternatively, both the easel output configurations 618, 620 may be presented simultaneously. In some embodiments, the easel output configurations 618, 620 may be similar or identical to the landscape output configurations 612, 616. The device 100 may also be configured to display one or both of the easel output configurations 618, 620 while in a modified easel state 316. It should be appreciated that simultaneous utilization of the easel output configurations 618, 620 may facilitate two-person games (e.g., Battleship®, chess, checkers, etc.), multi-user conferences where two or more users share the same device 100, and other applications. As can be appreciated, it may also be possible to transition the device 100 from the display of one or both configurations 618, 620 to any other configuration described herein depending upon which state the device 100 is moved.

[0105] FIG. 6I depicts yet another output configuration that may be accommodated while the device 100 is in an open portrait state 320. Specifically, the device 100 may be configured to present a single continuous image across both touch sensitive displays 110, 114 in a portrait configuration referred to herein as a Portrait-Max (PMax) configuration 624. In this configuration, data (e.g., a single image, application, window, icon, video, etc.) may be split and displayed partially on one of the touch sensitive displays while the other portion of the data is displayed on the other touch sensitive display. The Pmax configuration 624 may facilitate a larger display and/or better resolution for displaying a particular image on the device 100.

Similar to other output configurations, it may be possible to transition the device 100 from the Pmax configuration 624 to any other output configuration described herein depending upon which state the device 100 is moved.

[0106] FIG. 6J depicts still another output configuration that may be accommodated while the device 100 is in an open landscape state 348. Specifically, the device 100 may be configured to present a single continuous image across both touch sensitive displays 110, 114 in a landscape configuration referred to herein as a Landscape-Max (LMax) configuration 628. In this configuration, data (e.g., a single image, application, window, icon, video, etc.) may be split and displayed partially on one of the touch sensitive displays while the other portion of the data is displayed on the other touch sensitive display. The Lmax configuration 628 may facilitate a larger display and/or better resolution for displaying a particular image on the device 100. Similar to other output configurations, it may be possible to transition the device 100 from the Lmax configuration 628 to any other output configuration described herein depending upon which state the device 100 is moved.

[0107] The device 100 manages desktops and/or windows with at least one window stack 1700, 1728, as shown in Figs. 21A and 21B. A window stack 1700, 1728 is a logical arrangement of active and/or inactive windows for a multi-screen device. For example, the window stack 1700, 1728 may be logically similar to a deck of cards, where one or more windows or desktops are arranged in order, as shown in Figs. 21A and 21B. An active window is a window that is currently being displayed on at least one of the touch sensitive displays 110, 114. For example, windows 104 and 108 are active windows and are displayed on touch sensitive displays 110 and 114. An inactive window is a window that was opened and displayed but is now "behind" an active window and not being displayed. In embodiments, an inactive window may be for an application that is suspended, and thus, the window is not displaying active content. For example, windows 1712, 1716, 1720, and 1724 are inactive windows.

[0108] A window stack 1700, 1728 may have various arrangements or organizational structures. In the embodiment shown in Fig. 21A, the device 100 includes a first stack 1760 associated with a first touch sensitive display 110 and a second stack associated with a second touch sensitive display 114. Thus, each touch sensitive display 110, 114 can have an associated window stack 1760, 1764. These two window stacks 1760, 1764 may have different numbers of windows arranged in the respective stacks 1760, 1764. Further, the two window stacks 1760,

1764 can also be identified differently and managed separately. Thus, the first window stack 1760 can be arranged in order from a first window 1704 to a next window 1720 to a last window 1724 and finally to a desktop 1722, which, in embodiments, is at the "bottom" of the window stack 1760. In embodiments, the desktop 1722 is not always at the "bottom" as application windows can be arranged in the window stack below the desktop 1722, and the desktop 1722 can be brought to the "top" of a stack over other windows during a desktop reveal. Likewise, the second stack 1764 can be arranged from a first window 1708 to a next window 1712 to a last window 1716, and finally to a desktop 1718, which, in embodiments, is a single desktop area, with desktop 1722, under all the windows in both window stack 1760 and window stack 1764. A logical data structure for managing the two window stacks 1760, 1764 may be as described in conjunction with Fig. 18.

[0109] Another arrangement for a window stack 1728 is shown in Fig. 21B. In this embodiment, there is a single window stack 1728 for both touch sensitive displays 110, 114. Thus, the window stack 1728 is arranged from a desktop 1758 to a first window 1744 to a last window 1756. A window can be arranged in a position among all windows without an association to a specific touch sensitive display 110, 114. In this embodiment, a window is in the order of windows. Further, at least one window is identified as being active. For example, a single window may be rendered in two portions 1732 and 1736 that are displayed on the first touch sensitive screen 110 and the second touch sensitive screen 114. The single window may only occupy a single position in the window stack 1728 although it is displayed on both displays 110, 114.

[0110] Yet another arrangement of a window stack 1760 is shown in Figs. 21C through 21E. The window stack 1760 is shown in three "elevation" views. In Fig. 21C, the top of the window stack 1760 is shown. Two sides of the window stack 1760 are shown in Figs. 21D and 21E. In this embodiment, the window stack 1760 resembles a stack of bricks. The windows are stacked on each other. Looking from the top of the window stack 1760 in Fig, 21C, only the top most windows in the window stack 1760 are seen in different portions of the composite display 1764. The composite display 1764 represents a logical model for the entire display area of the device 100, which can include touch sensitive display 110 and touch sensitive display 114. A desktop 1786 or a window can occupy part or all of the composite display 1764.

[0111] In the embodiment shown, the desktop 1786 is the lowest display or "brick" in the window stack 1760. Thereupon, window 1 1782, window 2 1782, window 3 1768, and window 4 1770 are layered. Window 1 1782, window 3 1768, window 2 1782, and window 4 1770 only occupy a portion of the composite display 1764. Thus, another part of the stack 1760 includes window 8 1774 and windows 5 through 7 shown in section 1790. Only the top window in any portion of the composite display 1764 is actually rendered and displayed. Thus, as shown in the top view in Fig. 21C, window 4 1770, window 8 1774, and window 3 1768 are displayed as being at the top of the display in different portions of the window stack 1760. A window can be dimensioned to occupy only a portion of the composite display 1760 to "reveal" windows lower in the window stack 1760. For example, window 3 1768 is lower in the stack than both window 4 1770 and window 8 1774 but is still displayed. A logical data structure to manage the window stack can be as described in conjunction with Fig. 22.

[0112] When a new window is opened, the newly activated window is generally positioned at the top of the stack. However, where and how the window is positioned within the stack can be a function of the orientation of the device 100, the context of what programs, functions, software, etc. are being executed on the device 100, how the stack is positioned when the new window is opened, etc. To insert the window in the stack, the position in the stack for the window is determined and the touch sensitive display 110, 114 to which the window is associated may also be determined. With this information, a logical data structure for the window can be created and stored. When user interface or other events or tasks change the arrangement of windows, the window stack(s) can be changed to reflect the change in arrangement. It should be noted that these same concepts described above can be used to manage the one or more desktops for the device 100.

[0113] A logical data structure 1800 for managing the arrangement of windows or desktops in a window stack is shown in Fig. 22. The logical data structure 1800 can be any data structure used to store data whether an object, record, file, etc. The logical data structure 1800 can be stored in any type of database or data storage system, regardless of protocol or standard. In embodiments, the logical data structure 1800 includes one or more portions, fields, attributes, etc. that store data in a logical arrangement that allows for easy storage and retrieval of the information. Hereinafrer, these one or more portions, fields, attributes, etc. shall be described simply as fields. The fields can store data for a window identifier 1804, dimensions 1808, a

stack position identifier 1812, a display identifier 1816, and/or an active indicator 1820. Each window in a window stack can have an associated logical data structure 1800. While only a single logical data structure 1800 is shown in Fig. 22, there may be more or fewer logical data structures 1800 used with a window stack (based on the number of windows or desktops in the stack), as represented by ellipses 1824. Further, there may be more or fewer fields than those shown in Fig. 22, as represented by ellipses 1828.

[0114] A window identifier 1804 can include any identifier (ID) that uniquely identifies the associated window in relation to other windows in the window stack. The window identifier 1804 can be a globally unique identifier (GUID), a numeric ID, an alphanumeric ID, or other type of identifier. In embodiments, the window identifier 1804 can be one, two, or any number of digits based on the number of windows that can be opened. In alternative embodiments, the size of the window identifier 1804 may change based on the number of windows opened. While the window is open, the window identifier 1804 may be static and remain unchanged.

[0115] Dimensions 1808 can include dimensions for a window in the composite display 1760. For example, the dimensions 1808 can include coordinates for two or more corners of the window or may include one coordinate and dimensions for the width and height of the window. These dimensions 1808 can delineate what portion of the composite display 1760 the window may occupy, which may the entire composite display 1760 or only part of composite display 1760. For example, window 4 1770 may have dimensions 1880 that indicate that the window 1770 will occupy only part of the display area for composite display 1760, as shown in Figs. 21C through 21E. As windows are moved or inserted in the window stack, the dimensions 1808 may change.

[0116] A stack position identifier 1812 can be any identifier that can identify the position in the stack for the window or may be inferred from the window's control record within a data structure, such as a list or a stack. The stack position identifier 1812 can be a GUID, a numeric ID, an alphanumeric ID, or other type of identifier. Each window or desktop can include a stack position identifier 1812. For example, as shown in Fig. 21A, window 1 1704 in stack 1 1760 can have a stack position identifier 1812 of 1 identifying that window 1704 is the first window in the stack 1760 and the active window. Similarly, window 6 1724 can have a stack position identifier 1812 of 3 representing that window 1724 is the third window in the stack 1760. Window 2 1708 can also have a stack position identifier 1812 of 1 representing that window 1708 is the first

window in the second stack 1764. As shown in Fig. 21B, window 1 1744 can have a stack position identifier 1812 of 1, window 3, rendered in portions 1732 and 1736, can have a stack position identifier 1812 of 3, and window 6 1756 can have a stack position identifier 1812 of 6. Thus, depending on the type of stack, the stack position identifier 1812 can represent a window's location in the stack.

[0117] A display identifier 1816 can identify that the window or desktop is associated with a particular display, such as the first display 110 or the second display 114, or the composite display 1760 composed of both displays. While this display identifier 1816 may not be needed for a multi-stack system, as shown in Fig. 21A, the display identifier 1816 can indicate whether a window in the serial stack of Fig. 21B is displayed on a particular display. Thus, window 3 may have two portions 1732 and 1736 in Fig. 21B. The first portion 1732 may have a display identifier 1816 for the first display while the second portion 1736 may have a display identifier 1816 for the second display 114. However, in alternative embodiments, the window may have two display identifier 1816 that represent that the window is displayed on both of the displays 110, 114, or a display identifier 1816 identifying the composite display. In another alternate embodiment, the window may have a single display identifier 1816 to represent that the window is displayed on both of the displays 110, 114.

[0118] Similar to the display identifier 1816, an active indicator 1820 may not be needed with the dual stack system of Fig. 21A, as the window in stack position 1 is active and displayed. In the system of Fig. 21B, the active indicator 1820 can indicate which window(s) in the stack is being displayed. Thus, window 3 may have two portions 1732 and 1736 in Fig. 21. The first portion 1732 may have an active indicator 1820 while the second portion 1736 may also have an active indicator 1820. However, in alternative embodiments, window 3 may have a single active indicator 1820. The active indicator 1820 can be a simple flag or bit that represents that the window is active or displayed.

[0119] An embodiment of a method 1900 for creating a window stack is shown in Fig. 23. While a general order for the steps of the method 1900 is shown in Fig. 23. Generally, the method 1900 starts with a start operation 1904 and ends with an end operation 1928. The method 1900 can include more or fewer steps or can arrange the order of the steps differently than those shown in Fig. 23. The method 1900 can be executed as a set of computer-executable instructions executed by a computer system and encoded or stored on a computer readable

medium. Hereinafrer, the method 1900 shall be explained with reference to the systems, components, modules, software, data structures, user interfaces, etc. described in conjunction with Figs. 1-22.

[0120] A multi-screen device 100 can receive activation of a window, in step 1908. In embodiments, the multi-screen device 100 can receive activation of a window by receiving an input from the touch sensitive display 110 or 114, the configurable area 112 or 116, a gesture capture region 120 or 124, or some other hardware sensor operable to receive user interface inputs. The processor may execute the Task Management Module 540 may receive the input. The Task Management Module 540 can interpret the input as requesting an application task to be executed that will open a window in the window stack.

[0121] In embodiments, the Task Management Module 540 places the user interface interaction in the task stack 552 to be acted upon by the Display Configuration Module 568 of the Multi-Display Management Module 524. Further, the Task Management Module 540 waits for information from the Multi-Display Management Module 524 to send instructions to the Window Management Module 532 to create the window in the window stack.

[0122] The Multi-Display Management Module 524, upon receiving instruction from the Task Management Module 540, determines to which touch portion of the composite display 1760, the newly activated window should be associated, in step 1912. For example, window 4 1770 is associated with the a portion of the composite display 1764 In embodiments, the device state module 574 of the Multi-Display Management Module 524 may determine how the device is oriented or in what state the device is in, e.g., open, closed, portrait, etc. Further, the preferences module 572 and/or requirements module 580 may determine how the window is to be displayed. The gesture module 576 may determine the user's intentions about how the window is to be opened based on the type of gesture and the location of where the gesture is made.

[0123] The Display Configuration Module 568 may use the input from these modules and evaluate the current window stack 1760 to determine the best place and the best dimensions, based on a visibility algorithm, to open the window. Thus, the Display Configuration Module 568 determines the best place to put the window at the top of the window stack 1760, in step 1916. The visibility algorithm, in embodiments, determines for all portions of the composite display, which windows are at the top of the stack. For example, the visibility algorithm determines that window 3 1768, window 4 1770, and window 8 1774 are at the top of the stack

1760 as viewed in Figs. 21C through 21E. Upon determining where to open the window, the Display Configuration Module 568 can assign a display identifier 816 and possibly dimensions 808 to the window. The display identifier 816 and dimensions 808 can then be sent back to the Task Management Module 540. The Task Management Module 540 may then assign the window a stack position identifier 812 indicating the windows position at the top of the window stack.

[0124] In embodiments, the Task Management Module 540 sends the window stack information and instructions to render the window to the Window Management Module 532. The Window Management Module 532 and the Task Management Module 540 can create the logical data structure 800, in step 1924. Both the Task Management Module 540 and the Window Management Module 532 may create and manage copies of the window stack. These copies of the window stack can be synchronized or kept similar through communications between the Window Management Module 532 and the Task Management Module 540. Thus, the Window Management Module 532 and the Task Management Module 540, based on the information determined by the Multi-Display Management Module 524, can assign dimensions 808, a stack position identifier 812 (e.g., window 1 1782, window 4 1770, etc.), a display identifier 816 (e.g., touch sensitive display 1 110, touch sensitive display 2 114, composite display identifier, etc,), and an active indicator 820, which is generally always set when the window is at the "top" of the stack. The logical data structure 800 may then be stored by both the Window Management Module 532 and the Task Management Module 540. Further, the Window Management Module 532 and the Task Management Module 540may thereinafter manage the window stack and the logical data structure(s) 800.

[0125] FIG. 24 depicts a further window stacking configuration. A plurality of windows 1, 2, 3, 4, 5, 6, 7, and 8, whether from the same or different multi-screen or single-screen applications, are depicted. The touch sensitive display 110 currently has window 4 in the active display position while touch sensitive display 114 currently has window 5 in the active display position. The stack for touch sensitive display 110, from top to bottom, has window 4 in the active display position and windows 3, 2, and 1 positioned behind it. The stack for touch sensitive display 114, from top to bottom, has window 5 in the active display position and windows 6, 7, and 8 positioned behind it.

[0126] The desktops D1, D2, D3, D4, D5, and D6 are positioned behind the window stacks. The desktops can be viewed as a desktop stack different from the window stack. Viewed in this manner, the touch sensitive display 110 has a corresponding desktop stack comprising desktops D3, D2, and D1,with desktop D1 being at the bottom 2300 stack position and with desktop D3 at the top stack position being capable of being displayed with window 4 (depending on the window position and size (whether maximized or minimized), and the touch sensitive display 114 has a corresponding desktop stack has a corresponding desktop stack comprising desktops D4, D5, and D6,with desktop D6 being at the bottom 2304 stack position and with desktop D4 at the top stack position being capable of being displayed with window 5 (depending on the window position and size (whether maximized or minimized). Conceptually, the desktops can be viewed as a canvas divided, in this example, into six sections of which two can be displayed, at any one time, the touch sensitive displays 110, 114. When the device 100 is in the closed state, this conceptual model, in one configuration, persists. In this configuration, only one window and desktop stacks can be seen (that corresponding to the primary screen) but the other window and desktop stacks are virtual; that is, they are maintained in memory but cannot be seen because the secondary screen is not enabled.

[0127] Various window navigation configurations will now be discussed with reference to FIGS. 7-12 and 15-16. The various configurations depict examples relating to navigating among independent or hierarchically related displayed images. In the various examples, middleware 520, particularly one or more of the Multi-Display Management (MDM) class 524, a Surface Cache class 528, a Window Management class 532, an Activity Management class 536, and an Application Management class 540, independently or collectively, detect a stimulus, such as receipt of user input, particularly user gestures (step 1400 of FIG. 14), determine a current state of the device 100, display 110, 114 or screen 104, 108 (such as a state of frame buffers 548, activity stack 552, and/or event buffer 556), a context associated with the stimulus, and/or a sensed location of a gesture (step 1404), map one or more rule sets to the user device state, display state, screen state, stimulus context, and/or sensed gesture location to determine an appropriate functional result to be performed ,(step 1408), and perform or cause to be performed the functional result (step 1412).

[0128] Referring to FIGS. 8A-D, the primary and secondary touch sensitive displays 110 and 114 are in portrait display orientation and provide first and second desktops D1 and D2,

respectively (Fig. 8A). The first and second desktops form an ordered set of desktops. The secondary screen 108 detects a user gesture on the touch sensitive display 114. In one example, the user gesture 700 is a drag and, in another example, the user gesture is a flick. As shown in Fig. 8B, the second desktop D2 has moved in the direction indicated to an active display position (such as in a respective frame buffer or stack) or state for the primary screen and is displayed by the touch sensitive display 110, and the first desktop D1 has moved from an active to an inactive display position for the primary screen. A third desktop D3 has moved from an inactive to an active display position for the secondary screen and is now displayed by the touch sensitive display 114. Commonly, only a drag beyond approximately one-half of a display's or screen's width is considered to be a successful desktop navigation operation.

[0129] As further shown in Fig. 8C, the secondary screen 108 detects a further user gesture on the touch sensitive display 114. As shown in Fig. 8D, the third desktop D3 has moved in the direction indicated to an active display position for the primary screen and is displayed by the touch sensitive display 110, and the second desktop D2 has moved from an active display position to an inactive position for the primary screen. A fourth desktop D4 has moved from an inactive to an active display position for the secondary screen and is now displayed by the touch sensitive display 114.

[0130] Although not shown, the user could use an opposing gesture sensed in the touch sensitive display 110 to move a desktop from left-to-right. For example, in Fig. 8B the user could use a drag or flick gesture 700, 704 in the touch sensitive display 110 to cause the first desktop D1 to move from an inactive display position to an active display position for the touch sensitive display 110, thereby causing the second desktop D2 to move to an active position for the touch sensitive display 114 and the third desktop D3 to move from an active to an inactive display position for the secondary screen.

[0131] In one configuration shown in FIGS. 7A-B, gestures received by a gesture capture region 120 or 124 do not allow navigation between desktops in a desktop stack. Stated another way, gestures received by a touch sensitive display enable navigation through an ordered set or stack of desktops but not through an ordered set or stack of windows. As discussed below, gestures received by a gesture capture region enable navigation through the ordered set or stack of windows but not through the ordered set or stack of desktops.

[0132] Although drag and flick gestures are depicted in Fig. 7-8, it is to be understood that only one gesture is required and that other gestures may be employed.

[0133] Unlike desktop navigation, gestures received by a gesture capture region 120 or 124 are used to move windows from one screen to the other (e.g., from one active or inactive position to another). As discussed below, this rule applies when the device 100 is in either the closed or opened state.

[0134] Referring now to FIGS. 9A-D, a (minimized) first window 1 is displayed by the touch sensitive display 110, and a second desktop D2 by the touch sensitive display 114. In Fig. 9A, a gesture, such as a drag or flick gesture 700, 704 is received by a gesture capture region 120. In response and as shown in FIGS. 9B and 9C, the first window 1 moves in the direction indicated (as shown by the cross-thatched displayed image) from an active display position of the touch sensitive display 110 (and primary screen 104) to an active display position of the touch sensitive display 114 (and secondary screen 108). As shown by Fig. 9D, the first desktop D1 is uncovered by the first window 1 and moves from an inactive to an active display position of the touch sensitive display 110 (and primary screen 104), and the second desktop D2 is now covered by the first window 1 and moves from an active to an inactive display position for the secondary screen. The second desktop D2 could be returned to an active display position for the secondary screen by providing an opposing user gesture to the gesture capture region 124. This would return the first window 1 to the active display position of the primary screen, as shown in Fig. 9A.

[0135] In one configuration, a well is displayed ahead of the window as the window moves from the touch sensitive display 110 to touch sensitive display 114. The well is a displayed image that is unable to receive user input or provide output to the user. It typically, though not necessarily, is a substantially solid colored displayed image. The well expands or slides from behind the window and moves towards the target touch sensitive display. When and as the well expands or moves, a seam 800 between the first and second touch sensitive displays and their respective display images is dimmed out completely to reveal the well background. When the well has moved and occupies fully the target touch sensitive display (in this example touch sensitive display 114) so that the prior display image is completely dimmed out or covered by the well (the prior display image is, in the example, the second desktop 114), the first window 1 moves over to occupy the target touch sensitive display. Until that time, the first window 1

remains as the displayed image for the source touch sensitive display (in this example touch sensitive display 110). The well is an affordance generally used for multi-screen applications and not for single-screen applications.

[0136] Referring to FIGS. 10A-C, window navigation when the device 100 is in the closed state is depicted. In the following examples, an object stack includes first, second, and third windows 1, 2, 3. In Fig. 10A, the touch sensitive display 110 for the primary screen displays the first window 1. A gesture, shown as a drag 700 (though a flick 704 could be used), is received by the gesture capture region 120 for the primary screen 104. The gesture causes the window 1 to move to an inactive display position and the window 2 to move to an active display position on touch sensitive display 110.

[0137] With reference to Fig. 10B, the touch sensitive display 110 for the primary screen displays the second window 2. A gesture, shown as a drag 700 (though a flick 704 could be used), is received by the gesture capture region 120 for the primary screen 104. In response, the second window 2 moves in the direction indicated, causing the third window 3 to move into the active display position in the window stack, whereby the touch sensitive display 110 now displays the third window 3.

[0138] In Fig. 10C, the touch sensitive display 110 for the primary screen displays the third window 3. A gesture, shown as a drag 700 (though a flick 704 could be used), is received by the gesture capture region 120 for the primary screen 104. The user, by the gesture, attempts to move "beyond" the object stack for the primary screen 104 (not shown). The gesture causes the third window 3 (or a desktop at an end of the stack) to move partially off screen (to the lefr) followed by a rapid return, such as by a snap or rubber-band effect, to the primary display 104. This result communicates to the user that the window (or desktop) cannot be moved in the indicated direction as the end of the object stack has been reached. A similar result would have been realized in Fig. 10A had the drag gesture 700 been in the opposite direction.

[0139] Referring to FIGS. 11A-C, window navigation when the device 100 is in the open state is depicted. In the following examples, an object stack shown for the sake of simplicity as being common to both touch sensitive displays 110, 114 includes first desktop D1, (minimized) first and second windows 1, 2, and second desktop D2. In Fig. 11A, the touch sensitive display 110 for the primary screen displays the first desktop D1, and the touch sensitive display 114 for the primary screen displays the first window 1. A gesture, shown as a drag 700 (though a flick 704

could be used), is received by the gesture capture region 124 for the secondary screen 108 (or by the gesture capture region 120 for the primary screen 104). Referring to Fig. 11B, the gesture causes the first window 1 to move in the direction indicated, causing the first window 1 to move from the active display position for the secondary screen 108 to the active display position for the primary screen 104 and the first desktop D1 to an inactive display position for the primary screen 104. The second window 2 has moved from an inactive display position into the active display position for the secondary screen 108, whereby the touch sensitive display 114 now displays the second window 2.

[0140] With further reference to Fig. 11B, the gesture capture region 124 (or the gesture capture region 120 for the primary screen 104) receives a further gesture from the user, depicted as a drag gesture 700 (though a flick gesture 704 could be employed). Referring to Fig. 11C, the gesture causes the second window 2 to move in the direction indicated, causing the second window 2 to move into the active display position for the primary screen 104 and the first window 1 to an inactive display position for the primary screen 104. The second desktop D2 has moved from an inactive display position into the active display position for the secondary screen 108, whereby the touch sensitive display 114 now displays the second desktop D2.

[0141] In Fig. 11C, a gesture, shown as a drag 700 (though a flick 704 could be used), is received by the gesture capture region 124 for the primary screen 104 (or by the gesture capture region 120 for the primary screen 104). The user, by the gesture, attempts to move "beyond" the stack for the secondary screen 108. The gesture causes the second desktop 2 to move partially off screen (to the left) followed by a rapid return, such as by a snap or rubber-band effect, to the secondary screen 108. This result communicates to the user that the windows cannot be moved in the indicated direction as the end of the stack has been reached.

[0142] In the examples above, a gesture in the opposing direction would cause an opposite result. For example, in Fig. 11B, a drag in the opposite direction would cause the touch sensitive display 110 of the primary screen to display the second window 2, and the touch sensitive display 114 of the primary screen to display the second desktop D2.

[0143] In the examples above, it is to be understood that one or more other gesture(s) may accomplish the same functional result. The mapping of specific gestures to produce a specified result may be varied to produce a virtually endless array of possible relationships.

[0144] Navigation in landscape display orientation is, in one configuration, similar to navigating in portrait display orientation. One difference is the orientation of the displayed image stack, which may be oriented clockwise or counter-clockwise, depending on the direction of rotation of the device 100. Referring to FIGS. 16A-B, the device 100 is open and the primary and secondary touch sensitive displays 110 and 114 are in landscape display orientation and provide first and second desktops D1 and D2, respectively (Fig. 16A). The first and second desktops form an ordered set of desktops. The secondary screen 108 (or primary screen 104) detects a user gesture on the touch sensitive display 114. In one example, the user gesture 700 is a drag and, in another example, the user gesture is a flick. As shown in Fig. 16B, the second desktop D2 has moved in the direction indicated to an active display position (such as in a respective frame buffer or stack) or state for the primary screen and is displayed by the touch sensitive display 110, and the first desktop D1 has moved from an active to an inactive display position for the primary screen. A third desktop D3 has moved from an inactive to an active display position for the secondary screen and is now displayed by the touch sensitive display 114. Commonly, only a drag beyond approximately one-half of a display's or screen's height is considered to be a successful desktop navigation operation.

[0145] As further shown in Fig. 16B, the secondary screen 108 (or primary screen 104) detects a further user gesture on the touch sensitive display 114. As shown in Fig. 16C, the third desktop D3 has moved in the direction indicated to an active display position for the primary screen and is displayed by the touch sensitive display 110, and the second desktop D2 has moved from an active display position to an inactive position in a for the primary screen. A fourth desktop D4 has moved from an inactive to an active display position for the secondary screen and is now displayed by the touch sensitive display 114.

[0146] Although not shown, the user could use an opposing gesture sensed in the touch sensitive display 110 to move a desktop from top-to-bottom. For example, in Fig. 16B the user could use a downward drag or flick gesture 700, 704 in the touch sensitive display 110 to cause the first desktop D1 to move from an inactive display position to an active display position for the touch sensitive display 110, thereby causing the second desktop D2 to move to an active position for the touch sensitive display 114 and the third desktop D3 to move from an active to an inactive display position for the secondary screen.

[0147] A further example will be discussed with reference to FIGS. 12A-B. The object stack includes first and second windows 1, 2. First and second windows are controlled by separate single-screen applications. In other configurations, the first and second windows are controlled by a common multi-screen application or by different multi-screen and single-screen applications. In Fig. 12A, the primary screen 104 displays the first desktop D1, and the secondary screen 108 displays first window 1. A drag (or flick) gesture is received by the gesture capture region 120 or 124. In response and as shown in Fig. 12B, the primary screen 104 now displays the first window 1, and the secondary screen 108 now displays the second window 2. Stated another way, the first and second desktops D2 and D3 are in inactive display positions while the first and second windows are in active display positions.

[0148] Referring to FIGS. 15A-C, window navigation when the device 100 is in the closed state and landscape display orientation is depicted. In the following examples, the object stack includes first, second, and third windows 1, 2, 3. First, second, and third windows 1, 2, 3 are controlled by different single-screen applications. In other configurations, the first, second, and third windows are controlled by a common multi-screen application or by different multi-screen and single-screen applications. In Fig. 15A, the touch sensitive display 110 for the primary screen displays the first window 1. A gesture, shown as an upward drag 700 (though an upward flick 704 could be used), is received by the gesture capture region 120 for the primary screen 104. The gesture causes the window 1 to move to an inactive display position and window 2, the next window in the stack, to move to an active display position.

[0149] With reference to Fig. 15B, the touch sensitive display 110 for the primary screen now displays the second window 2. A gesture, shown as an upward drag 700 (though an upward flick 704 could be used), is received by the gesture capture region 120 for the primary screen 104. In response, the second window 2 moves in the direction indicated to an inactive display position, causing the third window 3 to move into the active display position in the window stack, whereby the touch sensitive display 110 now displays the third window 3 (Fig. 15C).

[0150] In Fig. 15C, a further gesture, particularly an upward drag 700 or flick 704, received by the gesture capture region 120 for the primary screen 104 would attempt to move "beyond" the object stack for the primary screen 104. The gesture would cause the third window 3 to move partially off screen (upwards) followed by a rapid return, such as by a snap or rubber-band effect,

to the primary display 104. This result communicates to the user that the windows cannot be moved into the indicated direction as the end of the object stack has been reached.

[0151] An example will now be discussed with reference to Fig. 13. A flick gesture 704 is received in step 1300 by one or more of a gesture capture region 120, 124 and a touch sensitive display 110, 114. In response, middleware 520, for each active touch sensitive display, moves, in step 1304, each currently displayed image from an active to an inactive display position or to an active display position of the other touch sensitive display. The logic then terminates in step 1308.

[0152] A series of examples will now be discussed to illustrate the use of a flick application to move simultaneously multiple windows under the common control of a multi-screen application. Prior to discussing the examples, the concept of a root view of a multi-screen application will be discussed. As shown in Fig. 17, first window 1.1 is a root window, second window 1.1.1 is a node window, and third window 1.1.1.1 is a leaf window. The root window 1.1 may be a top level view of the hierarchical application window sequence such that there is no parent window corresponding to the root window. The root window may be a parent to a node window. One or more node windows may be provided that are related as parent/children. A node window may also serve as a leaf window. By leaf window, it is meant that the leaf window has no corresponding node screen for which the leaf window is a parent. As such, the leaf window does not have any children node windows, Windows 1.1.1.1 and 1.1.2 are examples of leaf windows. The root view of a selected executing multi-screen application comprises all of the related windows depicted in Fig. 21.

[0153] Referring again to FIGS. 18A-C, the primary and secondary touch sensitive displays 110 and 114 are in portrait display orientation and display, in active display positions, the third window 1.1.1.1 and second desktop D2, respectively (Fig. 18A). The displayed images are part of ordered set(s) of currently or previously displayed images, with the third window 1.1.1.1 and second desktop D2 occupying active display positions and the first and second windows 1.1 and 1.1.1 and first desktop D1 occupying inactive display positions. Stated another way, the stack corresponding to the primary screen 104 has from top-to-bottom the third window 1.1.1.1, the second window 1.1.1, the first window 1.1, and the first desktop D1. The first, second, and third windows are controlled by a multi-screen application and not by separate single-screen applications. The first, second, and third windows are minimized as they occupy only one of the

primary and secondary screens 104 and 108. One or more of the windows may be maximized by a suitable gesture to occupy both of the primary and secondary screens 104 and 108.

[0154] While a window maximization transition and first displayed image operation is being performed, the primary screen 104 detects receipt of a gesture, typically a flick gesture and typically by the gesture capture region 124. A flick gesture 704 can move an entire root view of the windows of a multi-screen application from one touch sensitive display to another, thereby uncovering a desktop.

[0155] As shown in Fig. 18C, the root view (the hierarchically arranged windows from a common multi-screen application), namely first, second, and third windows 1.1, 1.1.1, and 1.1.1.1, previously positioned above the first desktop D1 on the primary screen 104, have now been moved to the secondary screen 108 to uncover the first desktop D1 (Fig. 18C). Stated another way, the stack corresponding to the secondary screen 104 has from top-to-bottom the third window 1.1.1.1, the second window 1.1.1, the first window 1.1, and the second desktop D2. In the absence of the flick gesture, the drag gesture would have moved the third window 1.1.1.1 to the secondary screen 108 to reveal the second window 1.1.1 in the primary screen 104 (view not shown).

[0156] Referring now to FIGS. 19A-C, the primary and secondary touch sensitive displays 110 and 114 are in portrait display orientation and display first window 1.1 and second desktop D2, respectively (Fig. 9A). The displayed images are part of ordered set(s) of currently or previously displayed images, with the first window 1.1 and second desktop D2 occupying active display positions and the second and third windows 1.1.1 and 1.1.1.1 and first desktop D1 occupying inactive display positions. Stated another way, the stack corresponding to the primary screen 104 has from top-to-bottom the first window 1.1, the second window 1.1.1, the third window 1.1.1.1, and the first desktop D1. The first, second, and third windows are controlled by a multi-screen application and not by separate single-screen applications. The windows are shown as being minimized to occupy one of the primary and secondary screens 104 and 108. In Fig. 19B, the primary screen 104, commonly the gesture capture region 120, detects a user gesture, particularly a flick, on the gesture capture region 120 to perform a displayed image operation. The middleware 520 determines that the root view (the hierarchically arranged windows from a common multi-screen application) is to be moved to the secondary screen 108 in accordance with flick rules. As shown in Fig. 19C, the root view, namely first, second, and third windows

1.1, 1.1.1, and 1.1.1.1, previously positioned above the first desktop D1 on the primary screen 104, have now been moved to the secondary screen 108 to uncover the first desktop D1 (Fig. 19C). In the absence of the flick gesture, a drag gesture would have moved the second window 1.1.1 to the secondary screen 108 to reveal the first window 1.1 in the primary screen 104 (view not shown).

[0157] The middleware logic for the flick operation will now be discussed with reference to Fig. 20. In step 2000, a flick gesture 704 is received in a gesture capture region 120, 124. The middleware determines, in step 2004, whether the window in the active display position on the source or origin touch sensitive display is controlled by a multi-screen or single-screen application. When the window is controlled by a multi-screen application, the middleware, in step 2008, moves the root view from the origin touch sensitive display to the target touch sensitive display while maintaining the ordering of the root view stack. When the window is controlled by a single-screen application, the middleware, in step 2012, moves only the window in the active display position in the origin touch sensitive display to the active display position in the target touch sensitive display.

[0158] The exemplary systems and methods of this disclosure have been described in relation to communication devices. However, to avoid unnecessarily obscuring the present disclosure, the preceding description omits a number of known structures and devices. This omission is not to be construed as a limitation of the scopes of the claims. Specific details are set forth to provide an understanding of the present disclosure. It should however be appreciated that the present disclosure may be practiced in a variety of ways beyond the specific detail set forth herein.

[0159] Furthermore, while the exemplary aspects, embodiments, and/or configurations illustrated herein show the various components of the system collocated, certain components of the system can be located remotely, at distant portions of a distributed network, such as a LAN and/or the Internet, or within a dedicated system. Thus, it should be appreciated, that the components of the system can be combined in to one or more devices, such as a communication device, or collocated on a particular node of a distributed network, such as an analog and/or digital telecommunications network, a packet-switch network, or a circuit-switched network. It will be appreciated from the preceding description, and for reasons of computational efficiency, that the components of the system can be arranged at any location within a distributed network of

components without affecting the operation of the system. For example, the various components can be located in a switch such as a PBX and media server, gateway, in one or more communications devices, at one or more users' premises, or some combination thereof. Similarly, one or more functional portions of the system could be distributed between a telecommunications device(s) and an associated computing device.

[0160] Furthermore, it should be appreciated that the various links connecting the elements can be wired or wireless links, or any combination thereof, or any other known or later developed element(s) that is capable of supplying and/or communicating data to and from the connected elements. These wired or wireless links can also be secure links and may be capable of communicating encrypted information. Transmission media used as links, for example, can be any suitable carrier for electrical signals, including coaxial cables, copper wire and fiber optics, and may take the form of acoustic or light waves, such as those generated during radio-wave and infra-red data communications.

[0161] Also, while the flowcharts have been discussed and illustrated in relation to a particular sequence of events, it should be appreciated that changes, additions, and omissions to this sequence can occur without materially affecting the operation of the disclosed embodiments, configuration, and aspects.

[0162] A number of variations and modifications of the disclosure can be used. It would be possible to provide for some features of the disclosure without providing others.

[0163] For example in one alternative embodiment, the desktops are moved by gestures received in a gesture capture region 120, 124, and windows are moved by gestures received in a touch sensitive display 110, 114.

[0164] In yet another embodiment, the systems and methods of this disclosure can be implemented in conjunction with a special purpose computer, a programmed microprocessor or microcontroller and peripheral integrated circuit element(s), an ASIC or other integrated circuit, a digital signal processor, a hard-wired electronic or logic circuit such as discrete element circuit, a programmable logic device or gate array such as PLD, PLA, FPGA, PAL, special purpose computer, any comparable means, or the like. In general, any device(s) or means capable of implementing the methodology illustrated herein can be used to implement the various aspects of this disclosure. Exemplary hardware that can be used for the disclosed embodiments, configurations and aspects includes computers, handheld devices, telephones (e.g., cellular,

Internet enabled, digital, analog, hybrids, and others), and other hardware known in the art. Some of these devices include processors (e.g., a single or multiple microprocessors), memory, nonvolatile storage, input devices, and output devices. Furthermore, alternative software implementations including, but not limited to, distributed processing or component/object distributed processing, parallel processing, or virtual machine processing can also be constructed to implement the methods described herein.

[0165] In yet another embodiment, the disclosed methods may be readily implemented in conjunction with software using object or object-oriented software development environments that provide portable source code that can be used on a variety of computer or workstation platforms. Alternatively, the disclosed system may be implemented partially or fully in hardware using standard logic circuits or VLSI design. Whether software or hardware is used to implement the systems in accordance with this disclosure is dependent on the speed and/or efficiency requirements of the system, the particular function, and the particular software or hardware systems or microprocessor or microcomputer systems being utilized.

[0166] In yet another embodiment, the disclosed methods may be partially implemented in software that can be stored on a storage medium, executed on programmed general-purpose computer with the cooperation of a controller and memory, a special purpose computer, a microprocessor, or the like. In these instances, the systems and methods of this disclosure can be implemented as program embedded on personal computer such as an applet, JAVA® or CGI script, as a resource residing on a server or computer workstation, as a routine embedded in a dedicated measurement system, system component, or the like. The system can also be implemented by physically incorporating the system and/or method into a software and/or hardware system.

[0167] Although the present disclosure describes components and functions implemented in the aspects, embodiments, and/or configurations with reference to particular standards and protocols, the aspects, embodiments, and/or configurations are not limited to such standards and protocols. Other similar standards and protocols not mentioned herein are in existence and are considered to be included in the present disclosure. Moreover, the standards and protocols mentioned herein and other similar standards and protocols not mentioned herein are periodically superseded by faster or more effective equivalents having essentially the same functions. Such

replacement standards and protocols having the same functions are considered equivalents included in the present disclosure.

[0168] The present disclosure, in various aspects, embodiments, and/or configurations, includes components, methods, processes, systems and/or apparatus substantially as depicted and described herein, including various aspects, embodiments, configurations embodiments, subcombinations, and/or subsets thereof. Those of skill in the art will understand how to make and use the disclosed aspects, embodiments, and/or configurations after understanding the present disclosure. The present disclosure, in various aspects, embodiments, and/or configurations, includes providing devices and processes in the absence of items not depicted and/or described herein or in various aspects, embodiments, and/or configurations hereof, including in the absence of such items as may have been used in previous devices or processes, e.g., for improving performance, achieving ease and\or reducing cost of implementation.

[0169] The foregoing discussion has been presented for purposes of illustration and description. The foregoing is not intended to limit the disclosure to the form or forms disclosed herein. In the foregoing Detailed Description for example, various features of the disclosure are grouped together in one or more aspects, embodiments, and/or configurations for the purpose of streamlining the disclosure. The features of the aspects, embodiments, and/or configurations of the disclosure may be combined in alternate aspects, embodiments, and/or configurations other than those discussed above. This method of disclosure is not to be interpreted as reflecting an intention that the claims require more features than are expressly recited in each claim. Rather, as the following claims reflect, inventive aspects lie in less than all features of a single foregoing disclosed aspect, embodiment, and/or configuration. Thus, the following claims are hereby incorporated into this Detailed Description, with each claim standing on its own as a separate preferred embodiment of the disclosure.

[0170] Moreover, though the description has included description of one or more aspects, embodiments, and/or configurations and certain variations and modifications, other variations, combinations, and modifications are within the scope of the disclosure, e.g., as may be within the skill and knowledge of those in the art, after understanding the present disclosure. It is intended to obtain rights which include alternative aspects, embodiments, and/or configurations to the extent permitted, including alternate, interchangeable and/or equivalent structures, functions, ranges or steps to those claimed, whether or not such alternate, interchangeable and/or equivalent

structures, functions, ranges or steps are disclosed herein, and without intending to publicly dedicate any patentable subject matter.

What is claimed is:

1. A method, comprising:

receiving, by at least one of gesture capture region and a touch sensitive display, a flick gesture;

in response, determining, by processor executable middleware, that an origin touch sensitive display comprises a plurality of windows controlled by a common multi-screen application; and

moving, by the processor executable middleware, the plurality of windows, substantially simultaneously, ftom the origin touch sensitive display to a target touch sensitive display.

2. The method of claim 1, wherein the plurality of windows are hierarchically related to one another.

3. The method of claim 1, wherein an ordering of the plurality of windows is maintained before and after movement to the target touch sensitive display.

4. The method of claim 1, wherein the flick gesture is received by the gesture capture region, and wherein the gesture capture region is unable to display any displayed image.

5. A non-transient computer readable medium comprising processor executable instructions operable to perform the steps of claim 1.

6. A dual display communication device, comprising:

a gesture capture region to receive a flick gesture;

a first touch sensitive display to receive a flick gesture and display displayed images, wherein the displayed images are at least one of a desktop and window of an application;

a second touch sensitive display to receive a flick gesture and display displayed images; and

middleware operable to perform at least one of the following operations:

receiving, by at least one of gesture capture region and a touch sensitive display, a flick gesture;

in response, determining that an origin touch sensitive display comprises a plurality of windows controlled by a common multi-screen application; and

moving the plurality of windows, substantially simultaneously, ftom the origin touch sensitive display to a target touch sensitive display.

7. The device of claim 6, wherein the plurality of windows are hierarchically related to one another.

8. The device of claim 6, wherein an ordering of the plurality of windows is maintained before and after movement to the target touch sensitive display.

9. The method of claim 1, wherein the flick gesture is received by the gesture capture region, and wherein the gesture capture region is unable to display any displayed image.

ABSTRACT

[0171] The disclosed method and device are directed to navigation, by a dual display communication device, through display objects.

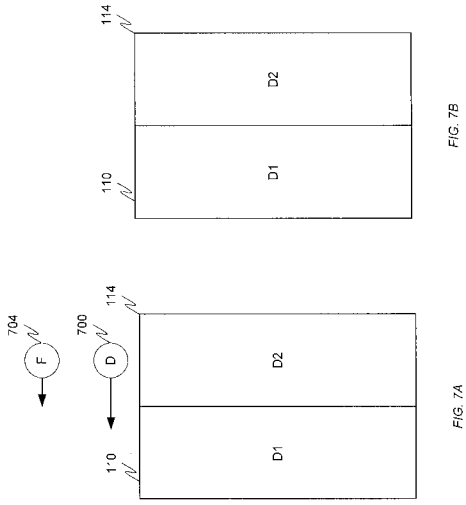


FIG. 7A

FIG. 7B

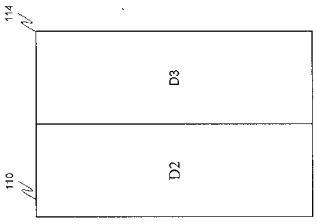


FIG. 8B

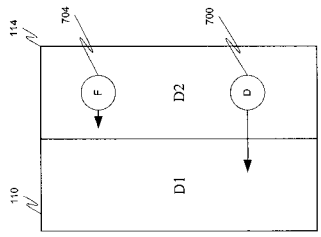


FIG. 8A

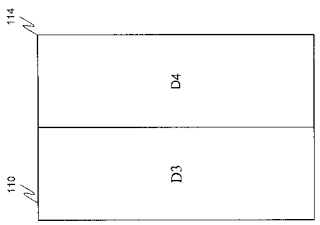


FIG. 8D

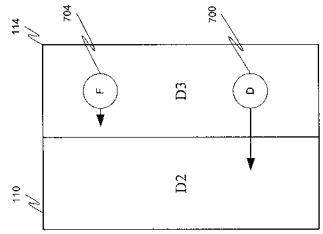


FIG. 8C

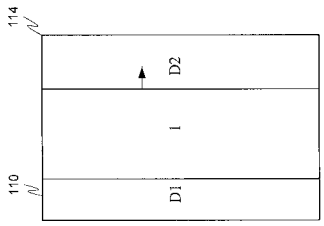


FIG. 9B

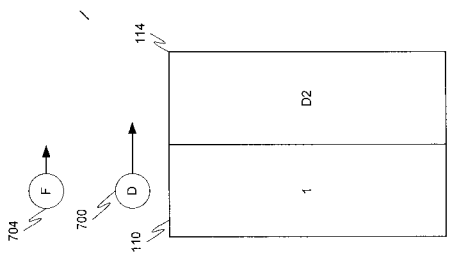


FIG. 9A

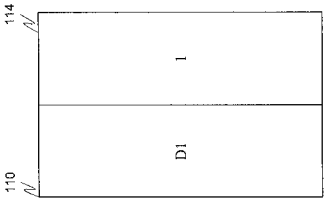


FIG. 9D

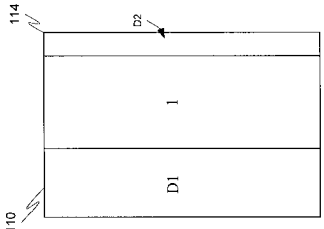


FIG. 9C

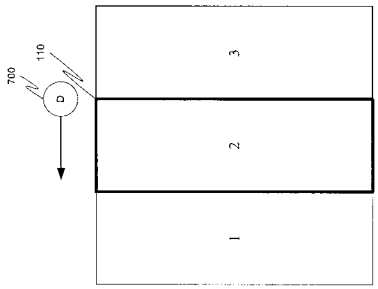


FIG. 10B

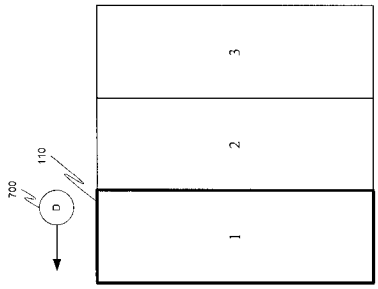


FIG. 10A

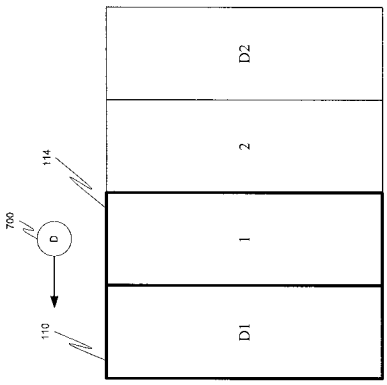


FIG. 11A

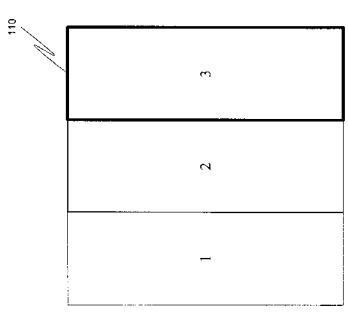


FIG. 10C

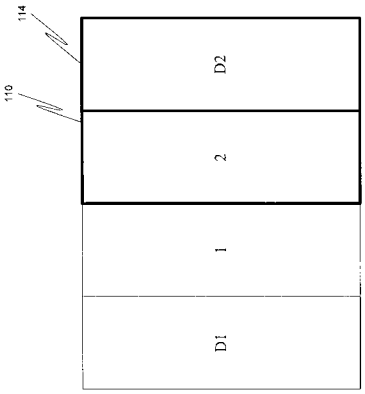


FIG. 11C

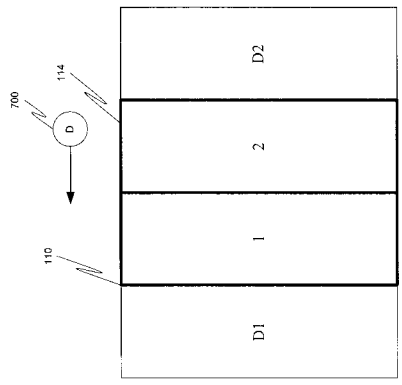


FIG. 11B

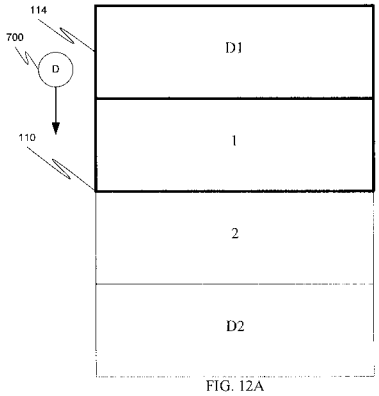


FIG. 12A

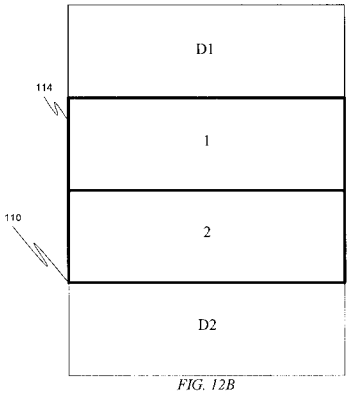


FIG. 12B

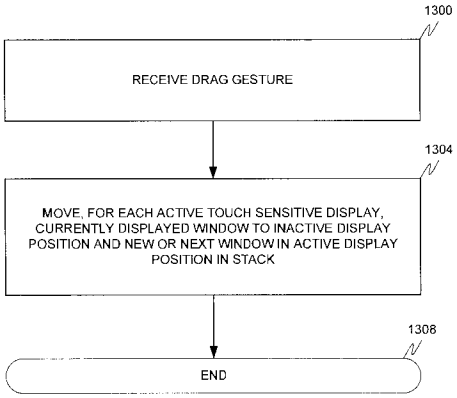


FIG. 13

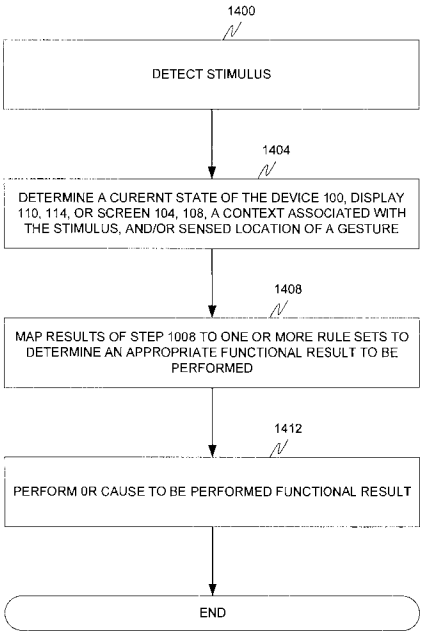


Fig. 14

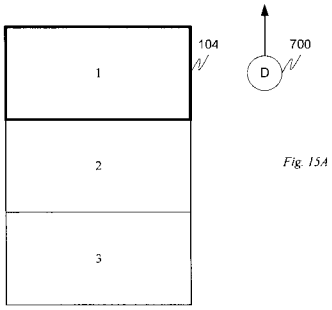


Fig. 15A

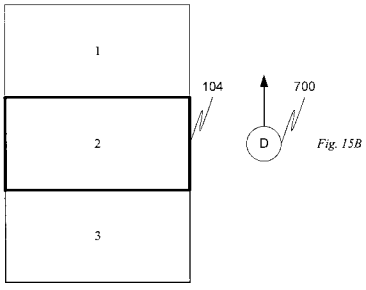


Fig. 15B

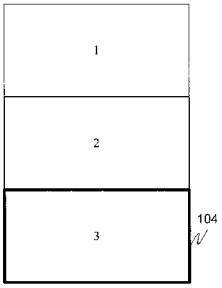


Fig. 15C

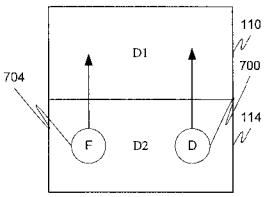


Fig. 16A

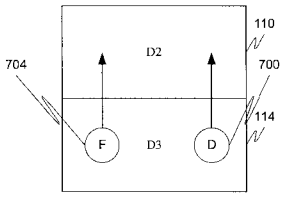


Fig. 16B

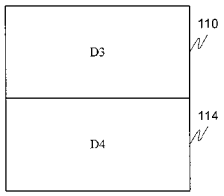


Fig. 16C

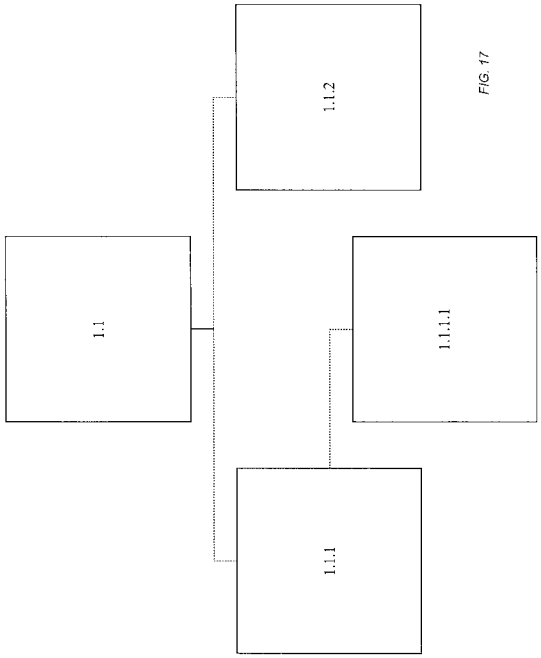


FIG. 17

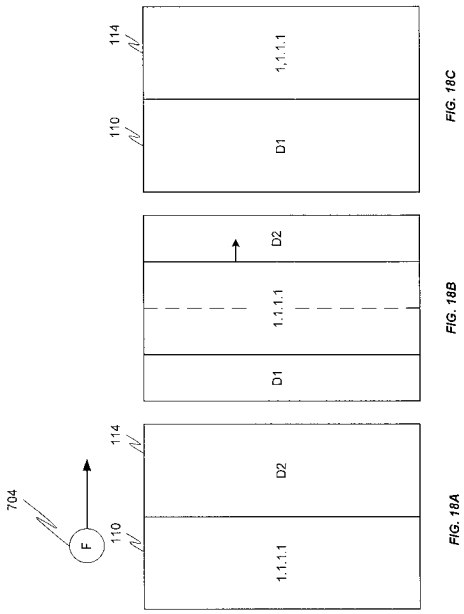


FIG. 18C

FIG. 18B

FIG. 18A

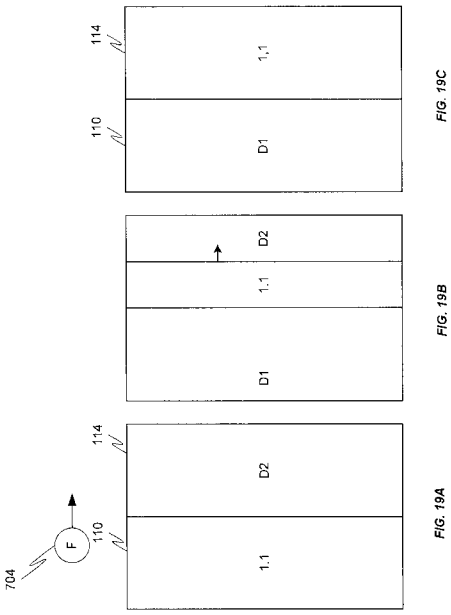


FIG. 19A

FIG. 19B

FIG. 19C

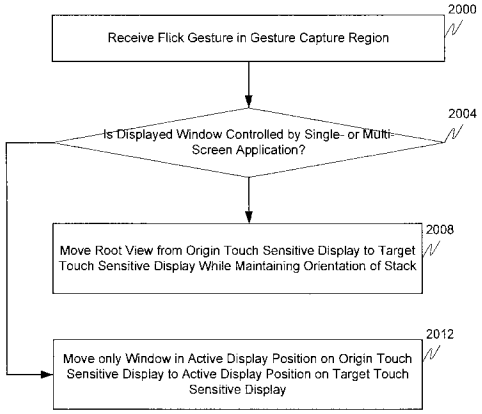


FIG. 20

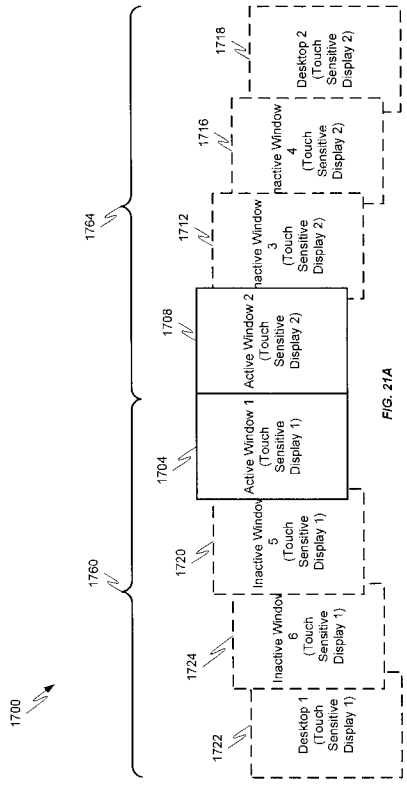


FIG. 21A

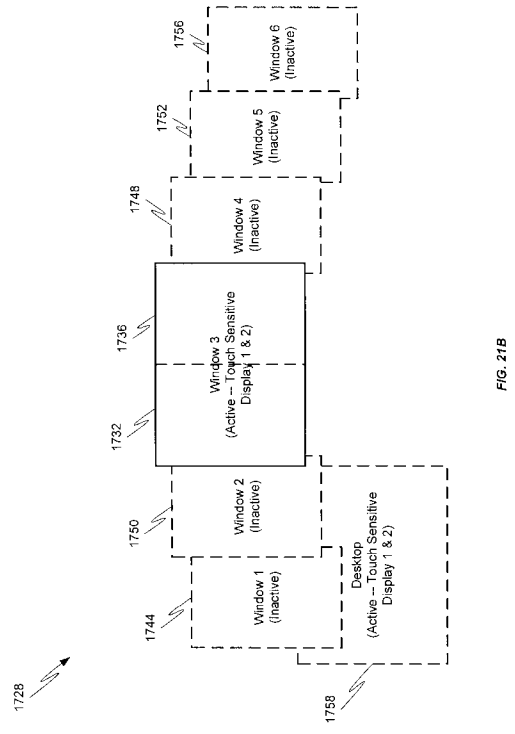


FIG. 21B

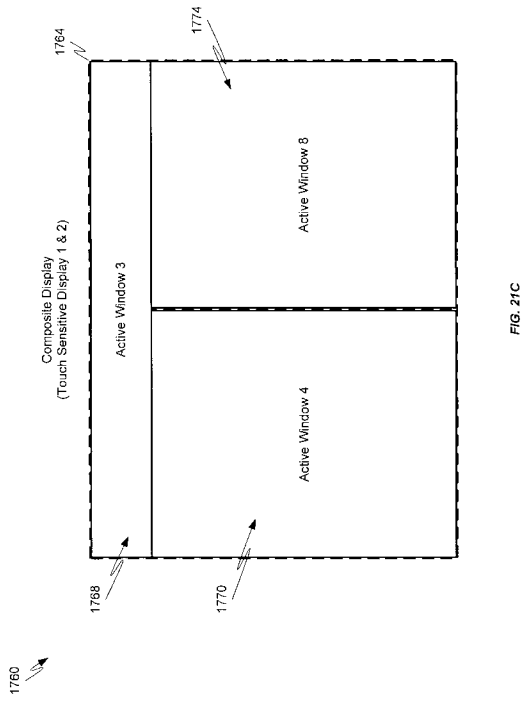


FIG. 21C

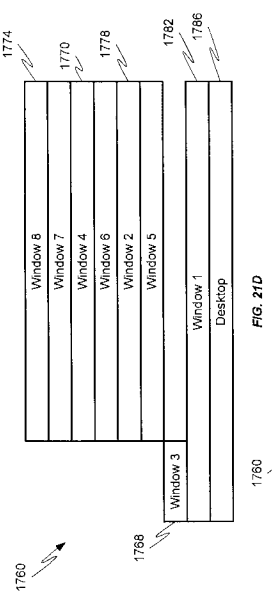


FIG. 21D

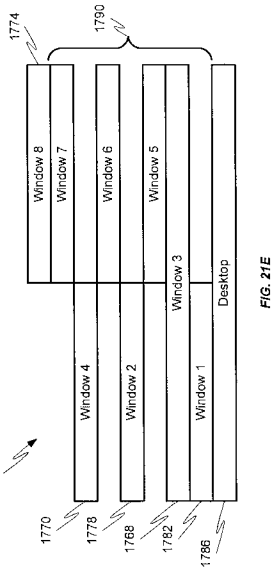


FIG. 21E

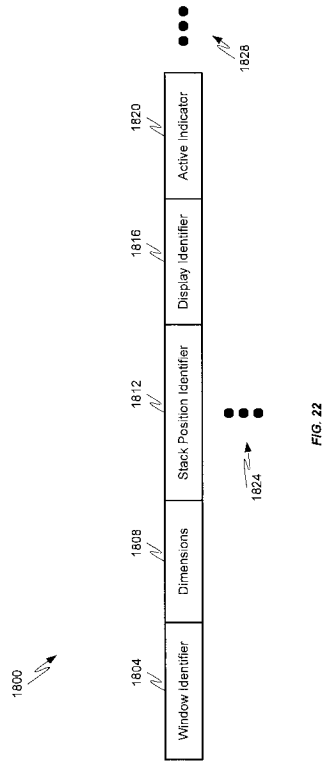


FIG. 22

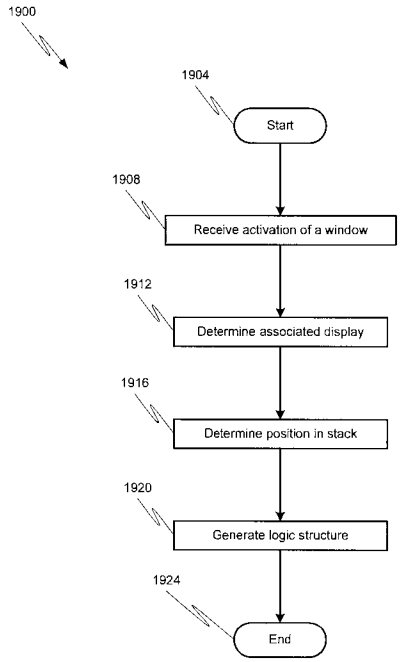


FIG. 23

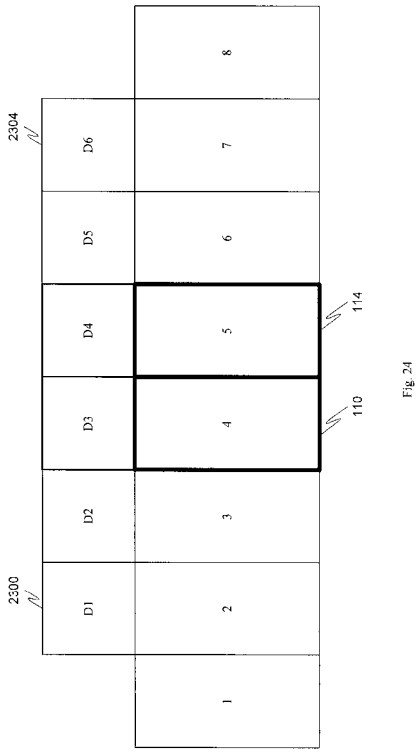


Fig. 24

DRAG MOVE GESTURE IN USER INTERFACE

[0100] The device 100 manages desktops and/or windows with at least one window stack 1700, 1728, as shown in Figs. 17A and 17B. A window stack 1700, 1728 is a logical arrangement of active and/or inactive windows for a multi-screen device. For example, the window stack 1700, 1728 may be logically similar to a deck of cards, where one or more windows or desktops are arranged in order, as shown in Figs. 17A and 17B. An active window is a window that is currently being displayed on at least one of the touch sensitive displays 110, 114. For example, windows 104 and 108 are active windows and are displayed on touch sensitive displays 110 and 114. An inactive window is a window that was opened and displayed but is now "behind" an active window and not being displayed. In embodiments, an inactive window may be for an application that is suspended, and thus, the window is not displaying active content. For example, windows 1712, 1716, 1720, and 1724 are inactive windows.

[0101] A window stack 1700, 1728 may have various arrangements or organizational structures. In the embodiment shown in Fig. 17A, the device 100 includes a first stack 1760 associated with a first touch sensitive display 110 and a second stack associated with a second touch sensitive display 114. Thus, each touch sensitive display 110, 114 can have an associated window stack 1760, 1764. These two window stacks 1760, 1764 may have different numbers of windows arranged in the respective stacks 1760, 1764. Further, the two window stacks 1760, 1764 can also be identified differently and managed separately. Thus, the first window stack 1760 can be arranged in order from a first window 1704 to a next window 1720 to a last window 1724 and finally to a desktop 1722, which, in embodiments, is at the "bottom" of the window stack 1760. In embodiments, the desktop 1722 is not always at the "bottom" as application windows can be arranged in the window stack below the desktop 1722, and the desktop 1722 can be brought to the "top" of a stack over other windows during a desktop reveal. Likewise, the second stack 1764 can be arranged from a first window 1708 to a next window 1712 to a last window 1716, and finally to a desktop 1718, which, in embodiments, is a single desktop area, with desktop 1722, under all the windows in both window stack 1760 and window stack 1764. A logical data structure for managing the two window stacks 1760, 1764 may be as described in conjunction with Fig. 18.

[0102] Another arrangement for a window stack 1728 is shown in Fig. 17B. In this embodiment, there is a single window stack 1728 for both touch sensitive displays 110, 114.

Thus, the window stack 1728 is arranged from a desktop 1758 to a first window 1744 to a last window 1756. A window can be arranged in a position among all windows without an association to a specific touch sensitive display 110, 114. In this embodiment, a window is in the order of windows. Further, at least one window is identified as being active. For example, a single window may be rendered in two portions 1732 and 1736 that are displayed on the first touch sensitive screen 110 and the second touch sensitive screen 114. The single window may only occupy a single position in the window stack 1728 although it is displayed on both displays 110, 114.

[0103] Yet another arrangement of a window stack 1760 is shown in Figs. 17C through 17E. The window stack 1760 is shown in three "elevation" views. In Fig. 17C, the top of the window stack 1760 is shown. Two sides of the window stack 1760 are shown in Figs. 17D and 17E. In this embodiment, the window stack 1760 resembles a stack of bricks. The windows are stacked on each other. Looking from the top of the window stack 1760 in Fig, 17C, only the top most windows in the window stack 1760 are seen in different portions of the composite display 1764. The composite display 1764 represents a logical model for the entire display area of the device 100, which can include touch sensitive display 110 and touch sensitive display 114. A desktop 1786 or a window can occupy part or all of the composite display 1764.

[0104] In the embodiment shown, the desktop 1786 is the lowest display or "brick" in the window stack 1760. Thereupon, window 1 1782, window 2 1782, window 3 1768, and window 4 1770 are layered. Window 1 1782, window 3 1768, window 2 1782, and window 4 1770 only occupy a portion of the composite display 1764. Thus, another part of the stack 1760 includes window 8 1774 and windows 5 through 7 shown in section 1790. Only the top window in any portion of the composite display 1764 is actually rendered and displayed. Thus, as shown in the top view in Fig. 17C, window 4 1770, window 8 1774, and window 3 1768 are displayed as being at the top of the display in different portions of the window stack 1760. A window can be dimensioned to occupy only a portion of the composite display 1760 to "reveal" windows lower in the window stack 1760. For example, window 3 1768 is lower in the stack than both window 4 1770 and window 8 1774 but is still displayed. A logical data structure to manage the window stack can be as described in conjunction with Fig. 18.

[0105] When a new window is opened, the newly activated window is generally positioned at the top of the stack. However, where and how the window is positioned within the stack can be

a function of the orientation of the device 100, the context of what programs, functions, software, etc. are being executed on the device 100, how the stack is positioned when the new window is opened, etc. To insert the window in the stack, the position in the stack for the window is determined and the touch sensitive display 110, 114 to which the window is associated may also be determined. With this information, a logical data structure for the window can be created and stored. When user interface or other events or tasks change the arrangement of windows, the window stack(s) can be changed to reflect the change in arrangement. It should be noted that these same concepts described above can be used to manage the one or more desktops for the device 100.

[0106] A logical data structure 1800 for managing the arrangement of windows or desktops in a window stack is shown in Fig. 18. The logical data structure 1800 can be any data structure used to store data whether an object, record, file, etc. The logical data structure 1800 can be stored in any type of database or data storage system, regardless of protocol or standard. In embodiments, the logical data structure 1800 includes one or more portions, fields, attributes, etc. that store data in a logical arrangement that allows for easy storage and retrieval of the information. Hereinafrer, these one or more portions, fields, attributes, etc. shall be described simply as fields. The fields can store data for a window identifier 1804, dimensions 1808, a stack position identifier 1812, a display identifier 1816, and/or an active indicator 1820. Each window in a window stack can have an associated logical data structure 1800. While only a single logical data structure 1800 is shown in Fig. 18, there may be more or fewer logical data structures 1800 used with a window stack (based on the number of windows or desktops in the stack), as represented by ellipses 1824. Further, there may be more or fewer fields than those shown in Fig. 18, as represented by ellipses 1828.

[0107] A window identifier 1804 can include any identifier (ID) that uniquely identifies the associated window in relation to other windows in the window stack. The window identifier 1804 can be a globally unique identifier (GUID), a numeric ID, an alphanumeric ID, or other type of identifier. In embodiments, the window identifier 1804 can be one, two, or any number of digits based on the number of windows that can be opened. In alternative embodiments, the size of the window identifier 1804 may change based on the number of windows opened. While the window is open, the window identifier 1804 may be static and remain unchanged.

[0108] Dimensions 1808 can include dimensions for a window in the composite display 1760. For example, the dimensions 1808 can include coordinates for two or more corners of the window or may include one coordinate and dimensions for the width and height of the window. These dimensions 1808 can delineate what portion of the composite display 1760 the window may occupy, which may the entire composite display 1760 or only part of composite display 1760. For example, window 4 1770 may have dimensions 1880 that indicate that the window 1770 will occupy only part of the display area for composite display 1760, as shown in Figs. 17C through 17E. As windows are moved or inserted in the window stack, the dimensions 1808 may change.

[0109] A stack position identifier 1812 can be any identifier that can identify the position in the stack for the window or may be inferred from the window's control record within a data structure, such as a list or a stack. The stack position identifier 1812 can be a GUID, a numeric ID, an alphanumeric ID, or other type of identifier. Each window or desktop can include a stack position identifier 1812. For example, as shown in Fig. 17A, window 1 1704 in stack 1 1760 can have a stack position identifier 1812 of 1 identifying that window 1704 is the first window in the stack 1760 and the active window. Similarly, window 6 1724 can have a stack position identifier 1812 of 3 representing that window 1724 is the third window in the stack 1760. Window 2 1708 can also have a stack position identifier 1812 of 1 representing that window 1708 is the first window in the second stack 1764. As shown in Fig. 17B, window 1 1744 can have a stack position identifier 1812 of 1, window 3, rendered in portions 1732 and 1736, can have a stack position identifier 1812 of 3, and window 6 1756 can have a stack position identifier 1812 of 6. Thus, depending on the type of stack, the stack position identifier 1812 can represent a window's location in the stack.

[0110] A display identifier 1816 can identify that the window or desktop is associated with a particular display, such as the first display 110 or the second display 114, or the composite display 1760 composed of both displays. While this display identifier 1816 may not be needed for a multi-stack system, as shown in Fig. 17A, the display identifier 1816 can indicate whether a window in the serial stack of Fig. 17B is displayed on a particular display. Thus, window 3 may have two portions 1732 and 1736 in Fig. 17B. The first portion 1732 may have a display identifier 1816 for the first display while the second portion 1736 may have a display identifier 1816 for the second display 114. However, in alternative embodiments, the window may have

two display identifier 1816 that represent that the window is displayed on both of the displays 110, 114, or a display identifier 1816 identifying the composite display. In another alternate embodiment, the window may have a single display identifier 1816 to represent that the window is displayed on both of the displays 110, 114.

[0111] Similar to the display identifier 1816, an active indicator 1820 may not be needed with the dual stack system of Fig. 17A, as the window in stack position 1 is active and displayed. In the system of Fig. 17B, the active indicator 1820 can indicate which window(s) in the stack is being displayed. Thus, window 3 may have two portions 1732 and 1736 in Fig. 17. The first portion 1732 may have an active indicator 1820 while the second portion 1736 may also have an active indicator 1820. However, in alternative embodiments, window 3 may have a single active indicator 1820. The active indicator 1820 can be a simple flag or bit that represents that the window is active or displayed.

[0112] An embodiment of a method 1900 for creating a window stack is shown in Fig. 19. While a general order for the steps of the method 1900 is shown in Fig. 19. Generally, the method 1900 starts with a start operation 1904 and ends with an end operation 1928. The method 1900 can include more or fewer steps or can arrange the order of the steps differently than those shown in Fig. 19. The method 1900 can be executed as a set of computer-executable instructions executed by a computer system and encoded or stored on a computer readable medium. Hereinafrer, the method 1900 shall be explained with reference to the systems, components, modules, software, data structures, user interfaces, etc. described in conjunction with Figs. 1-18.

[0113] A multi-screen device 100 can receive activation of a window, in step 1908. In embodiments, the multi-screen device 100 can receive activation of a window by receiving an input from the touch sensitive display 110 or 114, the configurable area 112 or 116, a gesture capture region 120 or 124, or some other hardware sensor operable to receive user interface inputs. The processor may execute the Task Management Module 540 may receive the input. The Task Management Module 540 can interpret the input as requesting an application task to be executed that will open a window in the window stack.

[0114] In embodiments, the Task Management Module 540 places the user interface interaction in the task stack 552 to be acted upon by the Display Configuration Module 568 of the Multi-Display Management Module 524. Further, the Task Management Module 540 waits

for information from the Multi-Display Management Module 524 to send instructions to the Window Management Module 532 to create the window in the window stack.

[0115] The Multi-Display Management Module 524, upon receiving instruction from the Task Management Module 540, determines to which touch portion of the composite display 1760, the newly activated window should be associated, in step 1912. For example, window 4 1770 is associated with the a portion of the composite display 1764 In embodiments, the device state module 574 of the Multi-Display Management Module 524 may determine how the device is oriented or in what state the device is in, e.g., open, closed, portrait, etc. Further, the preferences module 572 and/or requirements module 580 may determine how the window is to be displayed. The gesture module 576 may determine the user's intentions about how the window is to be opened based on the type of gesture and the location of where the gesture is made.

[0116] The Display Configuration Module 568 may use the input from these modules and evaluate the current window stack 1760 to determine the best place and the best dimensions, based on a visibility algorithm, to open the window. Thus, the Display Configuration Module 568 determines the best place to put the window at the top of the window stack 1760, in step 1916. The visibility algorithm, in embodiments, determines for all portions of the composite display, which windows are at the top of the stack. For example, the visibility algorithm determines that window 3 1768, window 4 1770, and window 8 1774 are at the top of the stack 1760 as viewed in Figs. 17C through 17E. Upon determining where to open the window, the Display Configuration Module 568 can assign a display identifier 816 and possibly dimensions 808 to the window. The display identifier 816 and dimensions 808 can then be sent back to the Task Management Module 540. The Task Management Module 540 may then assign the window a stack position identifier 812 indicating the windows position at the top of the window stack.

[0117] In embodiments, the Task Management Module 540 sends the window stack information and instructions to render the window to the Window Management Module 532. The Window Management Module 532 and the Task Management Module 540 can create the logical data structure 800, in step 1924. Both the Task Management Module 540 and the Window Management Module 532 may create and manage copies of the window stack. These copies of the window stack can be synchronized or kept similar through communications between the Window Management Module 532 and the Task Management Module 540. Thus,

the Window Management Module 532 and the Task Management Module 540, based on the information determined by the Multi-Display Management Module 524, can assign dimensions 808, a stack position identifier 812 (e.g., window 1 1782, window 4 1770, etc.), a display identifier 816 (e.g., touch sensitive display 1 110, touch sensitive display 2 114, composite display identifier, etc,), and an active indicator 820, which is generally always set when the window is at the "top" of the stack. The logical data structure 800 may then be stored by both the Window Management Module 532 and the Task Management Module 540. Further, the Window Management Module 532 and the Task Management Module 540may thereinafrer manage the window stack and the logical data structure(s) 800.

[0118] FIG. 20 depicts a further window stacking configuration. A plurality of windows 1, 2, 3, 4, 5, 6, 7, and 8, whether from the same or different multi-screen or single-screen applications, are depicted. The touch sensitive display 110 currently has window 4 in the active display position while touch sensitive display 114 currently has window 5 in the active display position. The stack for touch sensitive display 110, from top to bottom, has window 4 in the active display position and windows 3, 2, and 1 positioned behind it. The stack for touch sensitive display 114, from top to bottom, has window 5 in the active display position and windows 6, 7, and 8 positioned behind it.

[0119] The desktops D1, D2, D3, D4, D5, and D6 are positioned behind the window stacks. The desktops can be viewed as a desktop stack different from the window stack. Viewed in this manner, the touch sensitive display 110 has a corresponding desktop stack comprising desktops D3, D2, and D1,with desktop D1 being at the bottom 2300 stack position and with desktop D3 at the top stack position being capable of being displayed with window 4 (depending on the window position and size (whether maximized or minimized), and the touch sensitive display 114 has a corresponding desktop stack has a corresponding desktop stack comprising desktops D4, D5, and D6,with desktop D6 being at the bottom 2304 stack position and with desktop D4 at the top stack position being capable of being displayed with window 5 (depending on the window position and size (whether maximized or minimized). Conceptually, the desktops can be viewed as a canvas divided, in this example, into six sections of which two can be displayed, at any one time, the touch sensitive displays 110, 114. When the device 100 is in the closed state, this conceptual model, in one configuration, persists. In this configuration, only one window and desktop stacks can be seen (that corresponding to the primary screen) but the other

window and desktop stacks are virtual; that is, they are maintained in memory but cannot be seen because the secondary screen is not enabled.

[0120] Various window navigation configurations will now be discussed with reference to FIGS. 7-12 and 15-16. The various configurations depict examples relating to navigating among independent or hierarchically related displayed images. In the various examples, middleware 520, particularly one or more of the Multi-Display Management (MDM) class 524, a Surface Cache class 528, a Window Management class 532, an Activity Management class 536, and an Application Management class 540, independently or collectively, detect a stimulus, such as receipt of user input, particularly user gestures (step 1400 of FIG. 14), determine a current state of the device 100, display 110, 114 or screen 104, 108 (such as a state of frame buffers 548, activity stack 552, and/or event buffer 556), a context associated with the stimulus, and/or a sensed location of a gesture (step 1404), map one or more rule sets to the user device state, display state, screen state, stimulus context, and/or sensed gesture location to determine an appropriate functional result to be performed ,(step 1408), and perform or cause to be performed the functional result (step 1412).

[0121] Referring to FIGS. 8A-D, the primary and secondary touch sensitive displays 110 and 114 are in portrait display orientation and provide first and second desktops D1 and D2, respectively (Fig. 8A). The first and second desktops form an ordered set of desktops. The secondary screen 108 detects a user gesture on the touch sensitive display 114. In one example, the user gesture 700 is a drag and, in another example, the user gesture is a flick. As shown in Fig. 8B, the second desktop D2 has moved in the direction indicated to an active display position (such as in a respective frame buffer or stack) or state for the primary screen and is displayed by the touch sensitive display 110, and the first desktop D1 has moved from an active to an inactive display position for the primary screen. A third desktop D3 has moved from an inactive to an active display position for the secondary screen and is now displayed by the touch sensitive display 114. Commonly, only a drag beyond approximately one-half of a display's or screen's width is considered to be a successful desktop navigation operation.

[0122] As further shown in Fig. 8C, the secondary screen 108 detects a further user gesture on the touch sensitive display 114. As shown in Fig. 8D, the third desktop D3 has moved in the direction indicated to an active display position for the primary screen and is displayed by the touch sensitive display 110, and the second desktop D2 has moved from an active display

position to an inactive position for the primary screen. A fourth desktop D4 has moved from an inactive to an active display position for the secondary screen and is now displayed by the touch sensitive display 114.

[0123] Although not shown, the user could use an opposing gesture sensed in the touch sensitive display 110 to move a desktop from left-to-right. For example, in Fig. 8B the user could use a drag or flick gesture 700, 704 in the touch sensitive display 110 to cause the first desktop D1 to move from an inactive display position to an active display position for the touch sensitive display 110, thereby causing the second desktop D2 to move to an active position for the touch sensitive display 114 and the third desktop D3 to move from an active to an inactive display position for the secondary screen.

[0124] In one configuration shown in FIGS. 7A-B, gestures received by a gesture capture region 120 or 124 do not allow navigation between desktops in a desktop stack. Stated another way, gestures received by a touch sensitive display enable navigation through an ordered set or stack of desktops but not through an ordered set or stack of windows. As discussed below, gestures received by a gesture capture region enable navigation through the ordered set or stack of windows but not through the ordered set or stack of desktops.

[0125] Although drag and flick gestures are depicted in Fig. 7-8, it is to be understood that only one gesture is required and that other gestures may be employed.

[0126] Unlike desktop navigation, gestures received by a gesture capture region 120 or 124 are used to move windows from one screen to the other (e.g., from one active or inactive position to another). As discussed below, this rule applies when the device 100 is in either the closed or opened state.

[0127] Referring now to FIGS. 9A-D, a (minimized) first window 1 is displayed by the touch sensitive display 110, and a second desktop D2 by the touch sensitive display 114. The first window 1 is minimized in that it is displayed by only one of the primary and secondary screens 104 and 108. In the examples discussed herein, the windows and desktops can be maintained in separate display stacks or ordered sets. In Fig. 9A, a gesture, such as a drag or flick gesture 700, 704 is received by a gesture capture region 120. In response and as shown in FIGS. 9B and 9C, the first window 1 moves in the direction indicated (as shown by the cross-thatched displayed image) from an active display position of the touch sensitive display 110 (and primary screen 104) to an active display position of the touch sensitive display 114 (and secondary screen 108).

As shown by Fig. 9D, the first desktop D1 is uncovered by the first window 1 and moves from an inactive to an active display position of the touch sensitive display 110 (and primary screen 104), and the second desktop D2 is now covered by the first window 1 and moves from an active to an inactive display position for the secondary screen. The second desktop D2 could be returned to an active display position for the secondary screen by providing an opposing user gesture to the gesture capture region 124. This would return the first window 1 to the active display position of the primary screen, as shown in Fig. 9A.

[0128] In one configuration, a well is displayed ahead of the window as the window moves from the touch sensitive display 110 to touch sensitive display 114. The well is a displayed image that is unable to receive user input or provide output to the user. It typically, though not necessarily, is a substantially solid colored displayed image. The well expands or slides from behind the window and moves towards the target touch sensitive display. When and as the well expands or moves, a seam 800 between the first and second touch sensitive displays and their respective display images is dimmed out completely to reveal the well background. When the well has moved and occupies fully the target touch sensitive display (in this example touch sensitive display 114) so that the prior display image is completely dimmed out or covered by the well (the prior display image is, in the example, the second desktop 114), the first window 1 moves over to occupy the target touch sensitive display. Until that time, the first window 1 remains as the displayed image for the source touch sensitive display (in this example touch sensitive display 110). The well is an affordance generally used for multi-screen applications and not for single-screen applications.

[0129] Referring to FIGS. 10A-C, window navigation when the device 100 is in the closed state is depicted. In the following examples, an object stack includes first, second, and third windows 1, 2, 3. In Fig. 10A, the touch sensitive display 110 for the primary screen displays the first window 1. A gesture, shown as a drag 700 (though a flick 704 could be used), is received by the gesture capture region 120 for the primary screen 104. The gesture causes the window 1 to move to an inactive display position and the window 2 to move to an active display position on touch sensitive display 110.

[0130] With reference to Fig. 10B, the touch sensitive display 110 for the primary screen displays the second window 2. A gesture, shown as a drag 700 (though a flick 704 could be used), is received by the gesture capture region 120 for the primary screen 104. In response, the

second window 2 moves in the direction indicated, causing the third window 3 to move into the active display position in the window stack, whereby the touch sensitive display 110 now displays the third window 3.

[0131] In Fig. 10C, the touch sensitive display 110 for the primary screen displays the third window 3. A gesture, shown as a drag 700 (though a flick 704 could be used), is received by the gesture capture region 120 for the primary screen 104 (not shown). The user, by the gesture, attempts to move "beyond" the object stack for the primary screen 104. The gesture causes the third window 3 (or a desktop at an end of the stack) to move partially off screen (to the left) followed by a rapid return, such as by a snap or rubber-band effect, to the primary display 104. This result communicates to the user that the window (or desktop) cannot be moved in the indicated direction as the end of the object stack has been reached. A similar result would have been realized in Fig. 10A had the drag gesture 700 been in the opposite direction.

[0132] Referring to FIGS. 11A-C, window navigation when the device 100 is in the open state is depicted. In the following examples, an object stack shown for the sake of simplicity as being common to both touch sensitive displays 110, 114 includes first desktop D1, (minimized) first and second windows 1, 2, and second desktop D2. In Fig. 11A, the touch sensitive display 110 for the primary screen displays the first desktop D1, and the touch sensitive display 114 for the primary screen displays the first window 1. A gesture, shown as a drag 700 (though a flick 704 could be used), is received by the gesture capture region 124 for the secondary screen 108 (or by the gesture capture region 120 for the primary screen 104). Referring to Fig. 11B, the gesture causes the first window 1 to move in the direction indicated, causing the first window 1 to move from the active display position for the secondary screen 108 to the active display position for the primary screen 104 and the first desktop D1 to an inactive display position for the primary screen 104. The second window 2 has moved from an inactive display position into the active display position for the secondary screen 108, whereby the touch sensitive display 114 now displays the second window 2.

[0133] With further reference to Fig. 11B, the gesture capture region 124 (or the gesture capture region 120 for the primary screen 104) receives a further gesture from the user, depicted as a drag gesture 700 (though a flick gesture 704 could be employed). Referring to Fig. 11C, the gesture causes the second window 2 to move in the direction indicated, causing the second window 2 to move into the active display position for the primary screen 104 and the first

window 1 to an inactive display position for the primary screen 104. The second desktop D2 has moved from an inactive display position into the active display position for the secondary screen 108, whereby the touch sensitive display 114 now displays the second desktop D2.

[0134] In Fig. 11C, a gesture, shown as a drag 700 (though a flick 704 could be used), is received by the gesture capture region 124 for the primary screen 104 (or by the gesture capture region 120 for the primary screen 104). The user, by the gesture, attempts to move "beyond" the stack for the secondary screen 108. The gesture causes the second desktop 2 to move partially off screen (to the left) followed by a rapid return, such as by a snap or rubber-band effect, to the secondary screen 108. This result communicates to the user that the windows cannot be moved in the indicated direction as the end of the stack has been reached.

[0135] In the examples above, a gesture in the opposing direction would cause an opposite result. For example, in Fig. 11B, a drag in the opposite direction would cause the touch sensitive display 110 of the primary screen to display the second window 2, and the touch sensitive display 114 of the primary screen to display the second desktop D2.

[0136] In the examples above, it is to be understood that one or more other gesture(s) may accomplish the same functional result. The mapping of specific gestures to produce a specified result may be varied to produce a virtually endless array of possible relationships.

[0137] Navigation in landscape display orientation is, in one configuration, similar to navigating in portrait display orientation. One difference is the orientation of the displayed image stack, which may be oriented clockwise or counter-clockwise, depending on the direction of rotation of the device 100. Referring to FIGS. 16A-B, the device 100 is open and the primary and secondary touch sensitive displays 110 and 114 are in landscape display orientation and provide first and second desktops D1 and D2, respectively (Fig. 16A). The first and second desktops form an ordered set of desktops. The secondary screen 108 (or primary screen 104) detects a user gesture on the touch sensitive display 114. In one example, the user gesture 700 is a drag and, in another example, the user gesture is a flick. As shown in Fig. 16B, the second desktop D2 has moved in the direction indicated to an active display position (such as in a respective frame buffer or stack) or state for the primary screen and is displayed by the touch sensitive display 110, and the first desktop D1 has moved from an active to an inactive display position for the primary screen. A third desktop D3 has moved from an inactive to an active display position for the secondary screen and is now displayed by the touch sensitive display

114. Commonly, only a drag beyond approximately one-half of a display's or screen's height is considered to be a successful desktop navigation operation.

[0138] As further shown in Fig. 16B, the secondary screen 108 (or primary screen 104) detects a further user gesture on the touch sensitive display 114. As shown in Fig. 16C, the third desktop D3 has moved in the direction indicated to an active display position for the primary screen and is displayed by the touch sensitive display 110, and the second desktop D2 has moved from an active display position to an inactive position in a for the primary screen. A fourth desktop D4 has moved from an inactive to an active display position for the secondary screen and is now displayed by the touch sensitive display 114.

[0139] Although not shown, the user could use an opposing gesture sensed in the touch sensitive display 110 to move a desktop from top-to-bottom. For example, in Fig. 16B the user could use a downward drag or flick gesture 700, 704 in the touch sensitive display 110 to cause the first desktop D1 to move from an inactive display position to an active display position for the touch sensitive display 110, thereby causing the second desktop D2 to move to an active position for the touch sensitive display 114 and the third desktop D3 to move from an active to an inactive display position for the secondary screen.

[0140] A further example will be discussed with reference to FIGS. 12A-B. The object stack includes first and second windows 1, 2. First and second windows 1, 2 are controlled by different single-screen applications. In other configurations, the first and second windows are controlled by a common multi-screen application or by different multi-screen and single-screen applications. In Fig. 12A, the primary screen 104 displays the first desktop D1, and the secondary screen 108 displays first window 1. A drag (or flick) gesture is received by the gesture capture region 120 or 124. In response and as shown in Fig. 12B, the primary screen 104 now displays the first window 1, and the secondary screen 108 now displays the second window 2. Stated another way, the first and second desktops D2 and D3 are in inactive display positions while the first and second windows are in active display positions.

[0141] Referring to FIGS. 15A-C, window navigation when the device 100 is in the closed state and landscape display orientation is depicted. In the following examples, the object stack includes first, second, and third windows 1, 2, 3. First, second, and third windows 1, 2, 3 are controlled by different single-screen applications. In other configurations, the first, second, and third windows are controlled by a common multi-screen application or by different multi-screen

and single-screen applications. In Fig. 15A, the touch sensitive display 110 for the primary screen displays the first window 1. A gesture, shown as an upward drag 700 (though an upward flick 704 could be used), is received by the gesture capture region 120 for the primary screen 104. The gesture causes the window 1 to move to an inactive display position and window 2, the next window in the stack, to move to an active display position.

[0142] With reference to Fig. 15B, the touch sensitive display 110 for the primary screen now displays the second window 2. A gesture, shown as an upward drag 700 (though an upward flick 704 could be used), is received by the gesture capture region 120 for the primary screen 104. In response, the second window 2 moves in the direction indicated to an inactive display position, causing the third window 3 to move into the active display position in the window stack, whereby the touch sensitive display 110 now displays the third window 3 (Fig. 15C).

[0143] In Fig. 15C, a further gesture, particularly an upward drag 700 or flick 704, received by the gesture capture region 120 for the primary screen 104 would attempt to move "beyond" the object stack for the primary screen 104. The gesture would cause the third window 3 to move partially off screen (upwards) followed by a rapid return, such as by a snap or rubber-band effect, to the primary display 104. This result communicates to the user that the windows cannot be moved into the indicated direction as the end of the object stack has been reached.

[0144] An example will now be discussed with reference to Fig. 13. A drag gesture 700 is received in step 1300 by one or more of a gesture capture region 120, 124 and a touch sensitive display 110, 114. In response, middleware 520, for each active touch sensitive display, moves, in step 1304, each currently displayed image from an active to an inactive display position or to an active display position of the other touch sensitive display. The logic then terminates in step 1308.

[0145] The exemplary systems and methods of this disclosure have been described in relation to communication devices. However, to avoid unnecessarily obscuring the present disclosure, the preceding description omits a number of known structures and devices. This omission is not to be construed as a limitation of the scopes of the claims. Specific details are set forth to provide an understanding of the present disclosure. It should however be appreciated that the present disclosure may be practiced in a variety of ways beyond the specific detail set forth herein.

[0146] Furthermore, while the exemplary aspects, embodiments, and/or configurations illustrated herein show the various components of the system collocated, certain components of the system can be located remotely, at distant portions of a distributed network, such as a LAN and/or the Internet, or within a dedicated system. Thus, it should be appreciated, that the components of the system can be combined in to one or more devices, such as a communication device, or collocated on a particular node of a distributed network, such as an analog and/or digital telecommunications network, a packet-switch network, or a circuit-switched network. It will be appreciated from the preceding description, and for reasons of computational efficiency, that the components of the system can be arranged at any location within a distributed network of components without affecting the operation of the system. For example, the various components can be located in a switch such as a PBX and media server, gateway, in one or more communications devices, at one or more users' premises, or some combination thereof. Similarly, one or more functional portions of the system could be distributed between a telecommunications device(s) and an associated computing device.

[0147] Furthermore, it should be appreciated that the various links connecting the elements can be wired or wireless links, or any combination thereof, or any other known or later developed element(s) that is capable of supplying and/or communicating data to and from the connected elements. These wired or wireless links can also be secure links and may be capable of communicating encrypted information. Transmission media used as links, for example, can be any suitable carrier for electrical signals, including coaxial cables, copper wire and fiber optics, and may take the form of acoustic or light waves, such as those generated during radio-wave and infra-red data communications.

[0148] Also, while the flowcharts have been discussed and illustrated in relation to a particular sequence of events, it should be appreciated that changes, additions, and omissions to this sequence can occur without materially affecting the operation of the disclosed embodiments, configuration, and aspects.

[0149] A number of variations and modifications of the disclosure can be used. It would be possible to provide for some features of the disclosure without providing others.

[0150] For example in one alternative embodiment, the desktops are moved by gestures received in a gesture capture region 120, 124, and windows are moved by gestures received in a touch sensitive display 110, 114.

[0151] In yet another embodiment, the systems and methods of this disclosure can be implemented in conjunction with a special purpose computer, a programmed microprocessor or microcontroller and peripheral integrated circuit element(s), an ASIC or other integrated circuit, a digital signal processor, a hard-wired electronic or logic circuit such as discrete element circuit, a programmable logic device or gate array such as PLD, PLA, FPGA, PAL, special purpose computer, any comparable means, or the like. In general, any device(s) or means capable of implementing the methodology illustrated herein can be used to implement the various aspects of this disclosure. Exemplary hardware that can be used for the disclosed embodiments, configurations and aspects includes computers, handheld devices, telephones (e.g., cellular, Internet enabled, digital, analog, hybrids, and others), and other hardware known in the art. Some of these devices include processors (e.g., a single or multiple microprocessors), memory, nonvolatile storage, input devices, and output devices. Furthermore, alternative software implementations including, but not limited to, distributed processing or component/object distributed processing, parallel processing, or virtual machine processing can also be constructed to implement the methods described herein.

[0152] In yet another embodiment, the disclosed methods may be readily implemented in conjunction with software using object or object-oriented software development environments that provide portable source code that can be used on a variety of computer or workstation platforms. Alternatively, the disclosed system may be implemented partially or fully in hardware using standard logic circuits or VLSI design. Whether software or hardware is used to implement the systems in accordance with this disclosure is dependent on the speed and/or efficiency requirements of the system, the particular function, and the particular software or hardware systems or microprocessor or microcomputer systems being utilized.

[0153] In yet another embodiment, the disclosed methods may be partially implemented in software that can be stored on a storage medium, executed on programmed general-purpose computer with the cooperation of a controller and memory, a special purpose computer, a microprocessor, or the like. In these instances, the systems and methods of this disclosure can be implemented as program embedded on personal computer such as an applet, JAVA® or CGI script, as a resource residing on a server or computer workstation, as a routine embedded in a dedicated measurement system, system component, or the like. The system can also be

implemented by physically incorporating the system and/or method into a software and/or hardware system.

[0154] Although the present disclosure describes components and functions implemented in the aspects, embodiments, and/or configurations with reference to particular standards and protocols, the aspects, embodiments, and/or configurations are not limited to such standards and protocols. Other similar standards and protocols not mentioned herein are in existence and are considered to be included in the present disclosure. Moreover, the standards and protocols mentioned herein and other similar standards and protocols not mentioned herein are periodically superseded by faster or more effective equivalents having essentially the same functions. Such replacement standards and protocols having the same functions are considered equivalents included in the present disclosure.

[0155] The present disclosure, in various aspects, embodiments, and/or configurations, includes components, methods, processes, systems and/or apparatus substantially as depicted and described herein, including various aspects, embodiments, configurations embodiments, subcombinations, and/or subsets thereof. Those of skill in the art will understand how to make and use the disclosed aspects, embodiments, and/or configurations after understanding the present disclosure. The present disclosure, in various aspects, embodiments, and/or configurations, includes providing devices and processes in the absence of items not depicted and/or described herein or in various aspects, embodiments, and/or configurations hereof, including in the absence of such items as may have been used in previous devices or processes, e.g., for improving performance, achieving ease and\or reducing cost of implementation.

[0156] The foregoing discussion has been presented for purposes of illustration and description. The foregoing is not intended to limit the disclosure to the form or forms disclosed herein. In the foregoing Detailed Description for example, various features of the disclosure are grouped together in one or more aspects, embodiments, and/or configurations for the purpose of streamlining the disclosure. The features of the aspects, embodiments, and/or configurations of the disclosure may be combined in alternate aspects, embodiments, and/or configurations other than those discussed above. This method of disclosure is not to be interpreted as reflecting an intention that the claims require more features than are expressly recited in each claim. Rather, as the following claims reflect, inventive aspects lie in less than all features of a single foregoing disclosed aspect, embodiment, and/or configuration. Thus, the following claims are hereby

incorporated into this Detailed Description, with each claim standing on its own as a separate preferred embodiment of the disclosure.

[0157] Moreover, though the description has included description of one or more aspects, embodiments, and/or configurations and certain variations and modifications, other variations, combinations, and modifications are within the scope of the disclosure, e.g., as may be within the skill and knowledge of those in the art, after understanding the present disclosure. It is intended to obtain rights which include alternative aspects, embodiments, and/or configurations to the extent permitted, including alternate, interchangeable and/or equivalent structures, functions, ranges or steps to those claimed, whether or not such alternate, interchangeable and/or equivalent structures, functions, ranges or steps are disclosed herein, and without intending to publicly dedicate any patentable subject matter.

What is claimed is:

1. A method, comprising:

receiving, by at least one of a gesture capture region and a touch sensitive display, a gesture; and

in response, ceasing to display a first displayed image on a first touch sensitive display of a communication device and commencing to display the first displayed image on a second touch sensitive display of the communication device, wherein the first displayed image is at least one of a desktop and window of an application.

2. The method of claim 1, wherein the first displayed image is a desktop, wherein the gesture is received by the touch sensitive display, and wherein the desktop is one of a plurality of an ordered set of desktop images.

3. The method of claim 1, wherein the first displayed image is a window, wherein the gesture is received by the gesture capture region, wherein the gesture capture region is unable to display any displayed image, and wherein the window is one of a plurality of an ordered set of windows.

4. The method of claim 1, wherein a gesture received by the gesture capture region enables navigation through an ordered set of desktops or windows and a gesture received by the touch sensitive display enables navigation through an ordered set of the other of desktops or windows.

5. The method of claim 1, wherein the communication device has opened and closed states, wherein, in the opened state, both the first and second touch sensitive displays are enabled, wherein, in the closed state, the first touch sensitive display is enabled but the second touch sensitive display is disabled, and wherein the communication device is currently in the opened state.

6. The method of claim 1, wherein the received gesture is a drag gesture, wherein the first and second touch sensitive displays are each in the portrait display orientation, wherein the first touch sensitive display thereafter displays a second displayed image different from the first displayed image, wherein a third displayed image different ftom the first displayed image was previously displayed by the second touch sensitive display, and wherein the first and second displayed images, in response, are in active positions and the third displayed image is in an inactive position.

7. A non-transient computer readable medium comprising processor executable instructions operable to perform the steps of claim 1.

8. A method, comprising:

receiving, by at least one of a gesture capture region and a touch sensitive display, a gesture;

determining that a dual display communication device is in a closed state, wherein, in the closed state, a first touch sensitive display is enabled but a second touch sensitive display is disabled; and

in response, ceasing to display a first displayed image on the first touch sensitive display and displaying a different second displayed image on the first touch sensitive display, wherein the first displayed image is at least one of a desktop and window of an application.

9. The method of claim 8, wherein the first and second displayed images are first and second desktops, respectively, wherein the gesture is received by the touch sensitive display, and wherein the first and second desktops are adjacent members of an ordered set of desktop images.

10. The method of claim 8, wherein the first and second displayed images are first and second windows, respectively, wherein the gesture is received by the gesture capture region, wherein the gesture capture region is unable to display any displayed image, and wherein the first and second windows are adjacent members of an ordered set of windows.

11. The method of claim 8, wherein a gesture received by the gesture capture region enables navigation through an ordered set of desktops or windows and a gesture received by the touch sensitive display enables navigation through an ordered set of the other of desktops or windows.

12. The method of claim 8, wherein the received gesture is a drag gesture, wherein the first touch sensitive display is in the portrait display orientation, wherein the communication device has opened and closed states, wherein, in the opened state, both the first and second touch sensitive displays are enabled, wherein the first displayed image was previously in an active state and the second displayed image in an inactive state, and wherein, in response, the first displayed image is now in the inactive state and the second displayed image is now in the active state.

13. A non-transient computer readable medium comprising processor executable instructions operable to perform the steps of claim 8.

14. A dual display communication device, comprising:

a gesture capture region to receive a gesture;

a first touch sensitive display to receive a gesture and display displayed images, wherein the displayed images are at least one of a desktop and window of an application;

a second touch sensitive display to receive a gesture and display displayed images; and

middleware operable to perform at least one of the following operations:

in response to a received user gesture, andcommencing to display the first displayed image on the second touch sensitive display; and

determine that the dual display communication device is in a closed state, wherein, in the closed state, a first touch sensitive display is enabled but a second touch sensitive display is disabled and in response to the received user gesture, and displaying a different second displayed image on the first touch sensitive display.

15. The device of claim 14, wherein the middleware performs operation (A) and further comprising the operation of ceasing to display a first displayed image on the first touch sensitive display.

16. The device of claim 15, wherein the first displayed image is a desktop, wherein the gesture is received by the touch sensitive display, and wherein the desktop is one of a plurality of an ordered set of desktop images.

17. The device of claim 15, wherein the first displayed image is a window, wherein the gesture is received by the gesture capture region, wherein the gesture capture region is unable to display any displayed image, and wherein the window is one of a plurality of an ordered set of windows.

18. The device of claim 14, wherein the middleware performs operation (B) and further comprising the operation of ceasing to display the first displayed image on the first touch sensitive display.

19. The device of claim 18, wherein the first and second displayed images are first and second desktops, respectively, wherein the gesture is received by the touch sensitive display, and wherein the first and second desktops are adjacent members of an ordered set of desktop images.

20. The device of claim 18, wherein the first and second displayed images are first and second windows, respectively, wherein the gesture is received by the gesture capture region,

wherein the gesture capture region is unable to display any displayed image, and wherein the first and second windows are adjacent members of an ordered set of windows.

21. The device of claim 18, wherein the received gesture is a drag gesture, wherein the first touch sensitive display is in the portrait display orientation, wherein the communication device has opened and closed states, wherein, in the opened state, both the first and second touch sensitive displays are enabled, wherein the first displayed image was previously in an active state and the second displayed image in an inactive state, and wherein, in response, the first displayed image is now in the inactive state and the second displayed image is now in the active state.

22. The device of claim 14, wherein the received gesture is a drag gesture, wherein a gesture received by the gesture capture region enables navigation through an ordered set of desktops or windows and a gesture received by the touch sensitive display enables navigation through an ordered set of the other of desktops or windows.

ABSTRACT

[0158] The disclosed method and device are directed to navigation, by a dual display communication device, through display objects.

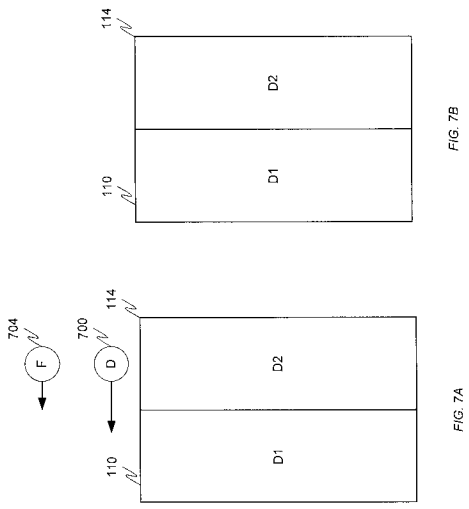


FIG. 7A

FIG. 7B

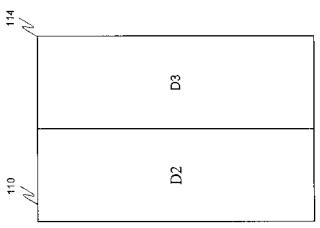


FIG. 8B

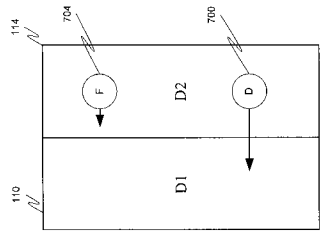


FIG. 8A

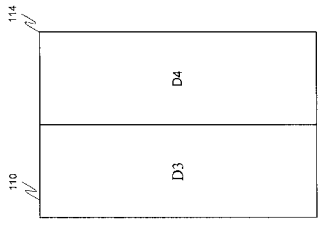


FIG. 8D

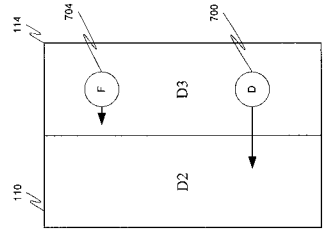


FIG. 8C

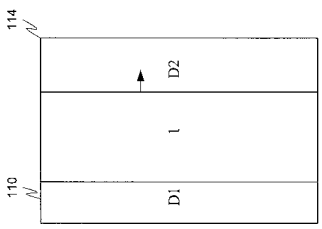


FIG. 9B

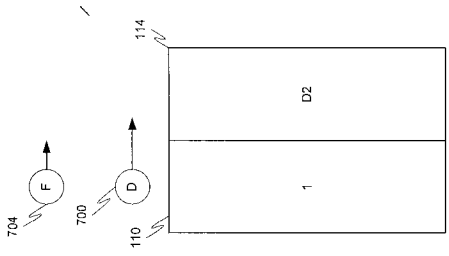


FIG. 9A

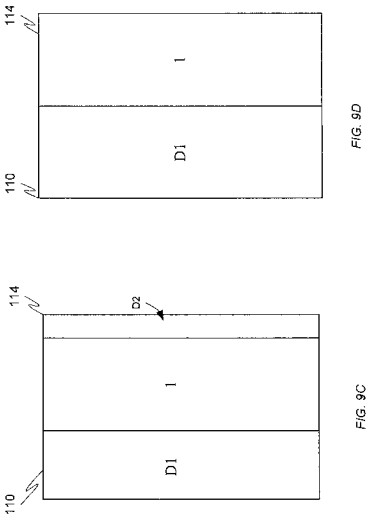


FIG. 9D

FIG. 9C

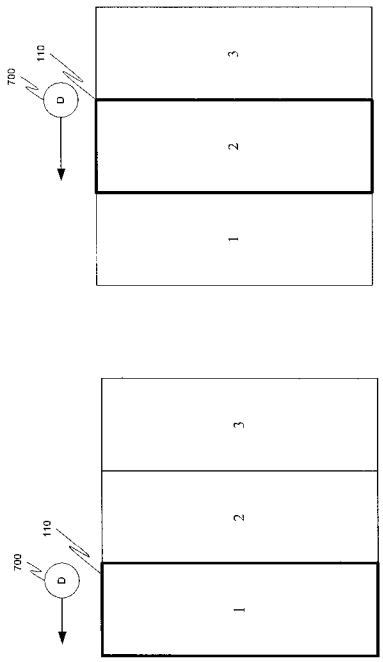


FIG. 10B

FIG. 10A

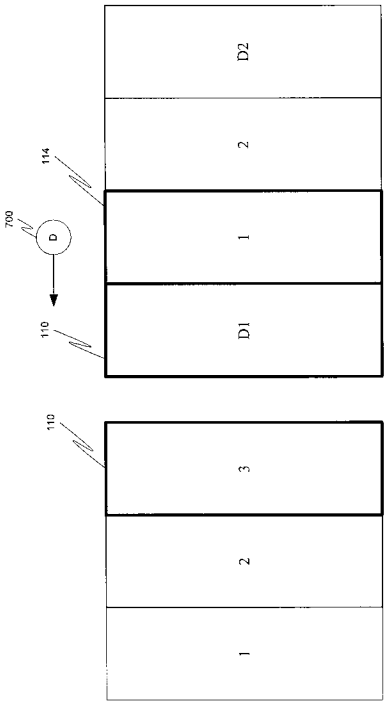


FIG. 11A

FIG. 10C

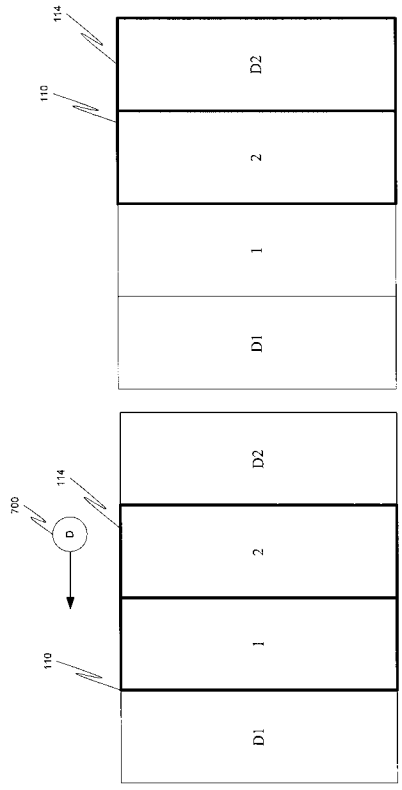


FIG. 11C

FIG. 11B

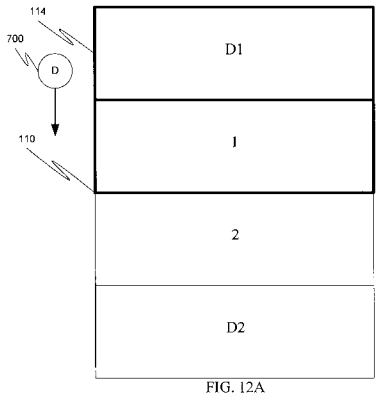


FIG. 12A

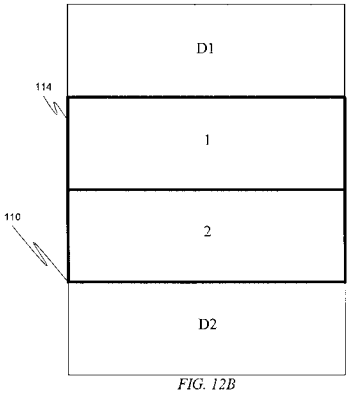


FIG. 12B

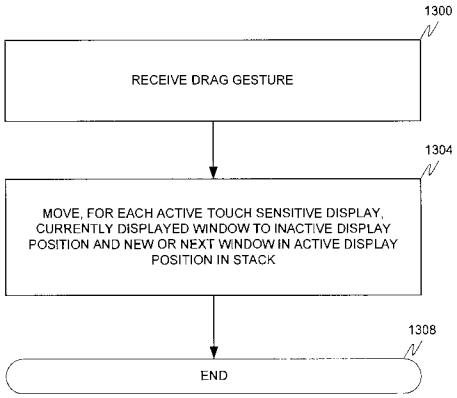


FIG. 13

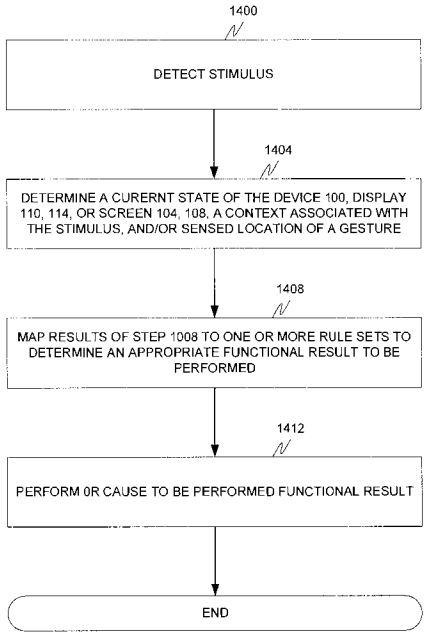


Fig. 14

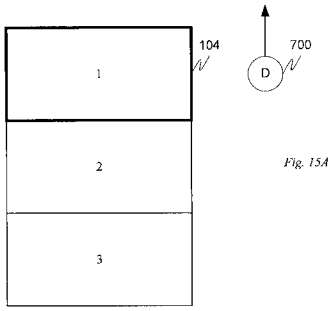


Fig. 15A

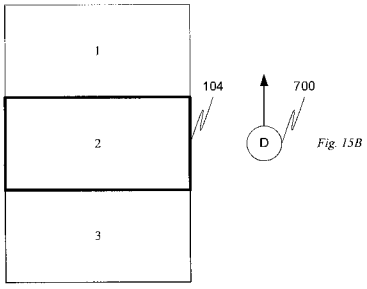


Fig. 15B

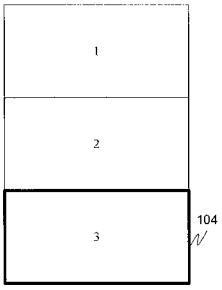


Fig. 15C

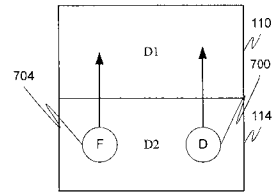


Fig. 16A

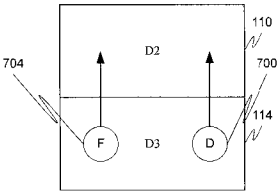


Fig. 16B

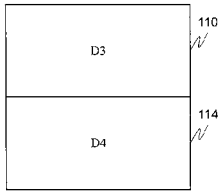


Fig. 16C

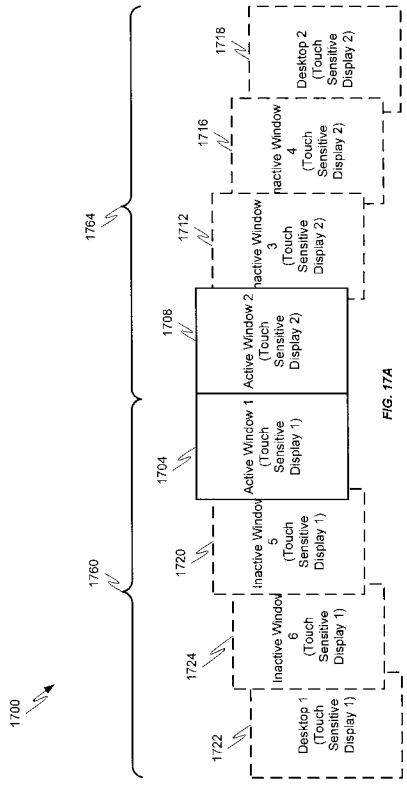


FIG. 17A

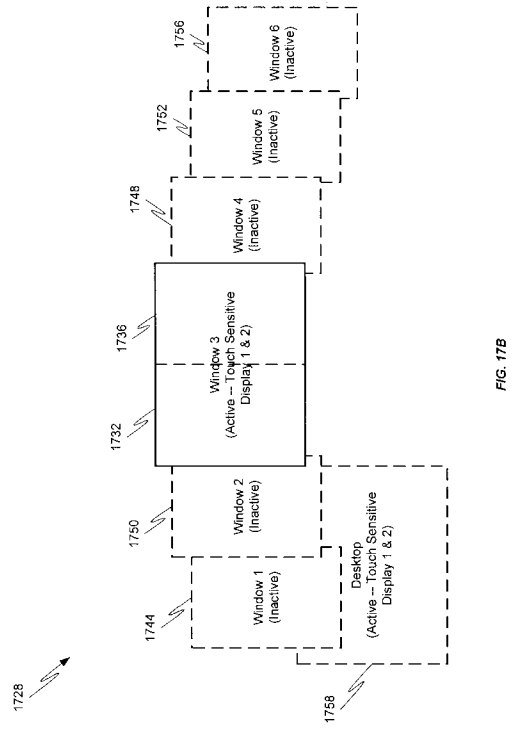


FIG. 17B

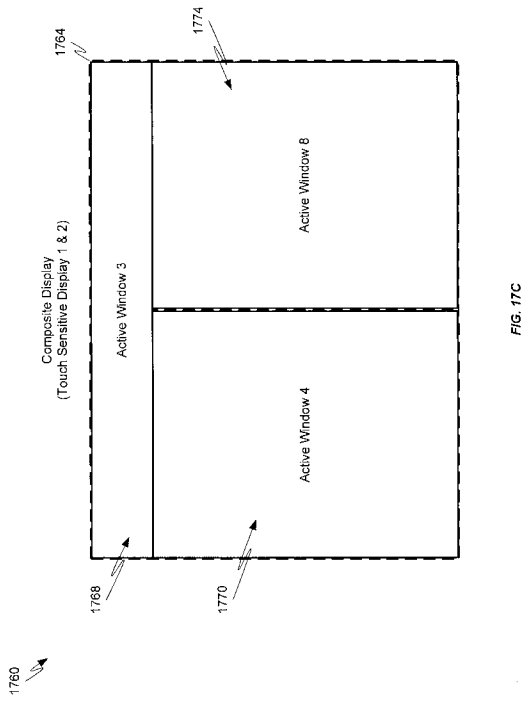


FIG. 17C

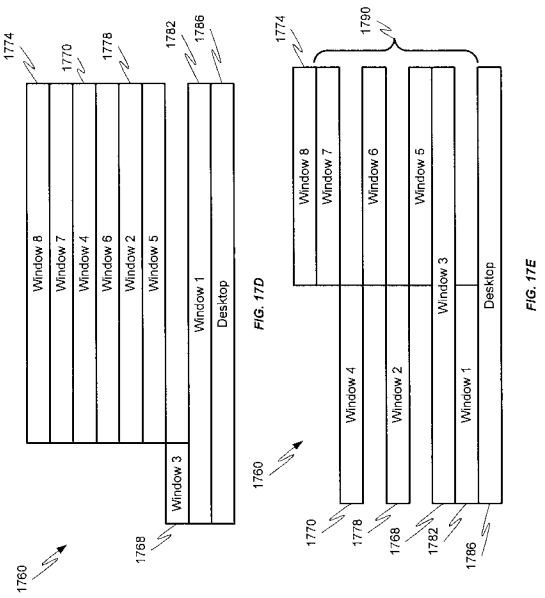


FIG. 17D

FIG. 17E

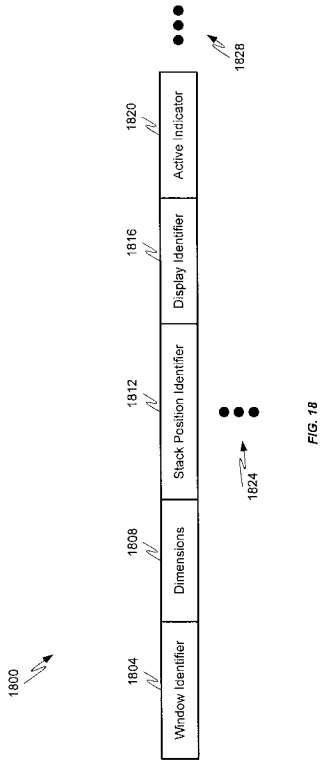


FIG. 18

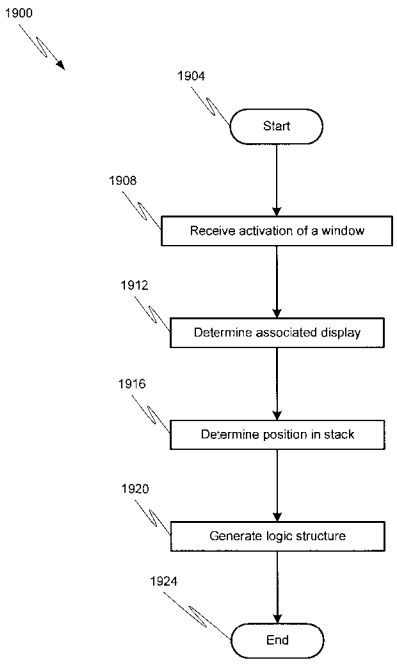


FIG. 19

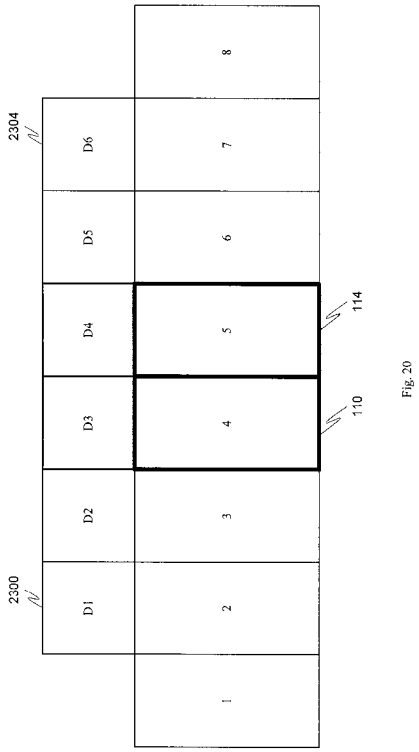


Fig. 20

[0001] Brief Description of the Drawings

[0002]

[0003] FIG. 10A depicts a first display state of an open device with a depiction of executing a spread gesture to launch an application manager;

[0004] FIG. 10B depicts a second display state of an open device with the application manager launched in the form of a separate window or tray that depicts the non-displayed applications and/or desktops running on the device, and also depicting the stack arrangement of the non-displayed applications and/or desktops; this Figure 10B also displays execution of a user gesture to rearrange the order of the window stack by a gesture executed on the primary screen;

[0005] FIG. 10C depicts a third display state of an open device after execution of the gesture in FIG. 10B , and also another spread gesture to re-launch the application manager;

[0006] FIG. 10D shows a fourth display state of the open device in which the application manager is launched and the non-displayed applications and/or desktops running on the device are shown in the manager window or tray in a new order based on the user gesture executed in FIG. 10B;

[0007] FIG. 10E depicts a first display state of an open device in another embodiment with a depiction of executing a spread gesture to launch an application manager;

[0008] FIG. 10F depicts a second display state of the open device of FIG. 10E with the application manager launched showing the manager window or tray that depicts the non-displayed applications and/or desktops running on the device, and also showing the stack arrangement of the non-displayed applications and/or desktops; this Figure 10F also displays execution of a user gesture to rearrange the order of the window stack by a gesture executed within the application manager to move a non-displayed application or desktop onto the primary screen;

[0009] FIG. 10G depicts a third display state of the open device of FIG. 10E after execution of the gesture in FIG 10F , and also another spread gesture to re-launch the application manager;

[0010] FIG. 10H shows a fourth display state of the open device of FIG 10E in which the application manager is launched and the non-displayed applications and/or desktops running on the device are shown in a new order based on the user gesture executed in FIG. 10F;

[0011] Figure 10I displays a first display state of an open device in yet another embodiment with the application manager launched showing the manager window or tray with multiple applications/desktops, and showing execution of a user gesture in the form of a drag gesture to rearrange the order of the window stack

[0012] FIG 10J displays a second display state of the open device of Figure 10I with the application manager launched showing the manager window or tray with the multiple applications/desktops, and showing execution of another user gesture in the form of a drag gesture to further rearrange the order of the window stack;

[0013] FIG 10K displays a third display state of the open device of Figure 10I with the application manager launched showing the manager window or tray with the multiple applications/desktops, and showing execution of yet another user gesture in the form of a tap gesture to further rearrange the order of the window stack in which the gesture opens the corresponding application/desktop in a predetermined screen;

[0014] FIG 10L displays a fourth display state of the open device of Figure 10I showing the new application displayed in the primary screen and the application manager closed; this Figure further depicts execution of another spread gesture to re-launch the application manager;

[0015] FIG. 10M displays a fifth display state of the open device of Figure 10I in which the manager window or tray shows the new window stack order responsive to the previous user gestures that rearranged the window stack; and

[0100] FIG. 11 shows a flow diagram depicting logic for executing rearrangement of a selected windo

[0101] Detailed Description

[0102]

[0103] Figure 10A depicts graphical representations of embodiments of a device 100 in an open portrait state where the primary screen is also visible as indicated by a solid line. The primary screen 104 shows a first page 1004 running a single page application (A5), and the secondary display 108 shows a first page 1006 running another single page application (A6). Fig. 10A also shows a user input gesture 1002, namely, a spread gesture which is executed in order to launch the application manager window. Referring to Figure 10B, the device 100 is shown after the spread gesture 1002 is executed ftom Figure 10A thereby revealing the application manager "pop up" window or tray 1008. Within this window 1008, depictions are shown for all of the other applications and desktops that are running at the time, but are not displayed on the primary or secondary screens. These depictions of each of the applications/desktops are shown by the icons 1010, each icon representing a separate application or desktop. The application manager window or tray 1008 also illustrates each of the other running applications/desktops in their respective order within the window stacks.

[0104] One convention that is contemplated in the preferred embodiment of the present invention for displaying an order of a window stack when the window 1008 is activated is to depict each of the other applications/desktops with their respective icons 1010 placed next to one another in a linear format ftom left to right, and in which the right-most icon 1010 represents an application/desktop which is the top or lead application/desktop within the management window 1008, and the left-most icon 1010 represents an application/desktop which is the lowest or bottom application/desktop in the window stack. The icons representing the application/windows between the left and right-most icons are depicted in their respective orders as compared to the top and bottom application/desktops. Therefore, in the example of Fig. 10B, application A1 is the bottom application, A2 is the next to the bottom application, application A3 is the next application in the stack above application A2, and application A4 represents the top or lead application in the group of applications illustrated in the window 1008. Fig. 10B also illustrates the manager window 1008 as being separated by line 1014. This line represents a division of the manager window 1008 and accordingly that both the primary and secondary screens have their own separate window stack. The application/desktops displayed on the primary and secondary screens are above or on top of the right-most applications/desktops displayed in the manager's window 1008. Therefore, applications A5 and A6 are the top applications in their respective window stacks. It is also contemplated that the manager window 1008 could represent a single stack common to both the primary and secondary screens. Thus, applications A5 andA6 would still be considered as applications residing above all of the others, but that one of the applications A5 or A6 would be the leading or very top application, and the other being the next application in the stack. Either application could be designated the leading or very top application in the single window stack according to the present invention. To designate a manager window with just one window stack for both screens, the line 1014 could be removed in accordance with yet another embodiment of the invention.

[0105] The shapes, sizes, and arrangements of the icons 1010 can be modified in order to provide the most convenient user interface enabling the user to view all of the running applications and desktops. In the example of Fig. 10B, the particular arrangement and size of the icons 1010 only allow three to be shown per screen. However, it is contemplated that the icons shown in Fig. 10B could be smaller or of different shapes enabling additional icons to be displayed if other applications were running. In the event there are more than three additional applications/desktops running on any of the screens, the bottommost non-displayed application/desktop would appear in the manager window if a user rearranged the window stack, and another application/desktop would become non-displayed.

[0106] Fig. 10B also illustrates another user gesture 1012, in the form of a drag gesture that allows the user to move the position of the selected application to the screen where the drag gesture is directed. Therefore referring to Fig. 10C, the state of the device 100 is shown after execution of the drag gesture 1012, and the application A5 is moved from the primary screen 104 to the secondary screen 108. Additionally, since the application A3 was the right-most application in the group of non-displayed applications/ desktops within the manager window 1008, this results in display of the application A3 on the primary screen 104. Accordingly the drag gesture, executed in Fig. 10B results in the display of the application A3 and movement of the application A5 ftom the primary screen to the secondary screen. The execution of the gesture 1012 causes the manager window 1008 to close.

[0107] Fig. 10C also illustrates another spread gesture 1002 that is executed to re-launch the application manager 1008. Accordingly, Fig. 10D illustrates the revised order for the application/desktops within the application manager 1008. More specifically, Fig. 10D shows application A3 on page 1016, application A5 is shown on page 1004, application A1 is shown on the window stack for the primary screen 104 as well as application A2, and applications A6 and A4 are shown on the window stack for the secondary screen 108. It is further noted with respect to Fig. 10D that the previously displayed application A6, after being "moved off" the secondary screen 108 by the movement of the application A5, now becomes the bottommost application in the window stack for the secondary screen 108.

[0108] While specific conventions are shown with respect to how the applications/desktops are arranged in an order within a particular window stack, it shall be understood that other conventions could be applied to include other methods of illustrating the particular order of each

of the applications/desktops within a window stacks. For example, instead of the right to left order within the window 1008 depicting a top to bottom arrangement of the applications or desktops, a left to right convention could be used indicating a top to bottom arrangement of the window/application. Further for example, the icons 1010 could appear in a manager window or tray that was oriented vertically on one of the screens, and the icons 1010 could appear in a column arrangement with the top or uppermost applications/desktop indicating the top applications/desktop in the stack.

[0109] Figures 10E-10H illustrate yet another embodiment of the present invention showing other ways in which a window stack may be manipulated by use of the application manager module/tool. Referring first to Fig. 10E, the device 100 in a first state shows page 1016 with application A3 and page 1006 with application A6. A spread gesture 1002 is executed. Now referring to Fig. 10F, the application manager window or tray 1008 is displayed and the user has executed yet another user gesture, such as a tap gesture 1020on a selected application/desktop. Tapping of one of the icons 1010 in this manner results in that particular application/desktop appearing on the screen ftom which the gesture was executed. Fig. 10F also illustrates the application A3 in the manager window1008 in dotted lines to indicate that that application will be moved from the primary screen 104 to a non-displayed application within the manager window 1008. Referring now to Fig. 10G, the tap gesture 1020 causes the manager window 1008 to close and the application A1 to appear on the primary screen 104 within another window 1022. Fig. 10G also illustrates the user executing another spread gesture 1002 in order to re-launch the application manager window. Therefore, referring to Fig. 10H a new order is illustrated for the window stack in which application A3 has been returned to the bottom of the window stack of the primary screen, and applications A4 and A5 still remain within the window stack of the secondary screen in the same order.

[0110] Fig. 10I -10M illustrate yet another preferred embodiment of the present invention. Referring first to Fig. 10I, the device 100 is shown after the user has launched the application manager, thereby displaying the window 1008 and the respective icons 1010. In the example of Fig. 10I, the user executes a drag gesture within the window 1008 in order to move the position of the selected application A5. More specifically, the user in Fig. 10I wishes to move the application A5 as the last application in the window stack for the secondary screen to the window stack for the primary screen 104.

[0111] Referring to Fig. 10J, it illustrates the application A5 as being moved to the stack within the primary screen and Fig. 10J also illustrates yet another drag gesture 1020 in which the user moves the application A4 into the window stack for the primary screen 104.

[0112] Now referring to Fig. 10K, this Figure illustrates no applications/desktops in the manager window for the secondary screen. For the primary screen, the applications A2, A5 and A6 are shown in the manager window 1008, but not the application A1. The application A1 is not illustrated because the particular size of the icons only allows a total of three applications/desktops to be displayed in the manager window per screen. Therefore, the application A1 is still running, but is simply not displayed in the manager window 1008. As mentioned above, if it were desired to also illustrate the application A1, the size, shape, or arrangement of the icons 1010 could be adjusted in order that all running applications/desktops could be illustrated within the application manager window 1008. Finally, Fig. 10K also illustrates the user executing a tap gesture 1020, which is the command to move the application A5 in a displayed position on the primary screen 104.

[0113] Now referring to Fig. 10L, the tap gesture 1020 closes the application manager window 1008, the application A5 is displayed on page 1004 on the primary screen 104, and application A6 in window 1006 remains displayed on the secondary screen 1008.

[0114] Fig. 10L also illustrates yet another spread gesture 1002, and in response as shown in Fig. 10M, the application manager window 1008 is again displayed, along with each of the icons 1010 illustrating the updated order of the window stack. Accordingly, applications A5 and A6 are displayed on the primary and secondary screens, respectively, no applications appear in the window stack for the secondary screen, and applications A1, A3, and A4 are illustrated in the manager window 1008 of the primary screen.

[0115] Now referring to Fig. 11, a simplified flow chart 1050 is provided to explain a basic sequence of events when a user launches the application manager in order to change the window stack(s) order. At the start 1052, the user determines the desired application/desktops to run, and then launches the applications, shown at block 1054. The user then must determine whether the apps are in the desired stack order, shown at block 1056. If the user believes the applications are in the desired stack order, the analysis is complete, shown at end block 1070. If the applications are not in the desired order, the user then executes a gesture to initiate the app manager, shown at block 1058. The user then views the applications as they are represented by the icons 1010 within the manager window 1008, and the user also determines a desired order, shown at block 1060. The user then executes the appropriate gesture to move a first selected application to a new position, shown at block 1062. In response to the executed gesture, the stack order changes, shown at block 1064. As discussed above with reference to Figs. 10A-10M, a user can execute a number of different gestures to move the applications into the desired order, such as a drag gesture on the screens 104 and 108, or tap or drag gestures within the application manager window 1008. At block 1068, The user then re-evaluates the applications in the stack and determines whether they are in the desired stack order after execution of the first move gesture, shown at block 1068. If the applications are now in the desired stack order, then the analysis is complete. If the applications are not yet in the desired stack order, the user then executes a second gesture to move another application to another location within the stack. The gestures are repeated until the user is satisfied with the desired window stack(s) order(s).

[0116] Based on the description of Figs. 10A-10M, it is evident that the display of windows in a window stack can be selectively changed by simple gesture interactions, thereby making desktops and multiple applications available to the user with minimum effort. Each screen has a logical representation of the windows that have been opened and displayed on the screen. The logical representation is the stack that is made up of one or more windows that is arranged in an order that can be selected and changed by the user. By opening of the window manager tool, the windows can be indexed or shifted to place the window stack in a desired order as well as to move windows between separate window stacks, such as separate window stacks that may be present on both the primary and secondary screens. The ordered arrangement of the stack maintains the logical construction of the stack intact when the user navigates the window stack(s). The user can open the manager window 1008 repeatedly in order to continue to change the order of the window stack(s), and each subsequent gesture executed is based on the present order of the window stack(s) prior to executing the subsequent gesture(s). Therefore, a logical order is maintained for each window stack at all times. The use of the icons 1010 within the window manager provides a very convenient visual representation of the running applications/desktops. Therefore the device operates in a very intuitive way for the user to easily manage multiple desktops and applications within a dual screen environment.

Claims

1. A method of displaying a plurality of desktops and/or applications, each having at least one window, on a multi-screen device, comprising:

receiving, by a processor, a first input that represents an instruction to reveal one of a desktop or application on a first display of the multi-screen device;

selecting, by a processor, a first desktop or application to display on the first display;

receiving, by a processor , a second input that represents an instruction to reveal one of a desktop or application on a second display of the multi-screen device;

selecting, by a processor, a second desktop or application to display on the second display;

displaying, by a processor, the selected first and second desktops or applications on the first and second displays;

receiving, by a processor, a third input that represents an instruction to launch an application manager feature;

displaying, by a processor, the applications or desktops on the first and second displays, and displaying a management window showing all other applications or desktops currently running on the device;

determining, by a processor, a desired order for a window stack comprising the first and second applications or desktops, and said other applications or desktops;

receiving, by a processor, a fourth input that represents an instruction to selectively move one of said first and second applications or desktops, or said other applications or desktops, within the window stack to a different position; and

displaying, by a processor, the window stack in a new order in response to said fourth input.

2. The method of claim 1, wherein the fourth input is a user gesture comprising at least one of a flick, tap, long hold, or drag gesture.

3. The method of claim 1, wherein the third input is a user gesture comprising a spread gesture, and wherein the spread gesture is executed on off-screen areas of said first and

seconds displays with one finger of a user placed on one screen, and another finger of the user placed on the other screen.

4. The method of claim 1, wherein said receiving a fourth input step is executed a number of times until the window stack is displayed in the desired order.

5. The method of claim 1, wherein at least one management window appears on said first and second displays, and the other applications or desktops are arranged linearly within the management window ftom left to right, and the left to right order of the other applications or desktops corresponds to an order of the other applications or desktops within the window stack.

6. The method of claim 5, wherein the rightmost appearing one of the other applications or desktops is the top application or desktop in the group of the other applications or desktops, and the leftmost appearing one of the other applications or desktops is the bottom application or desktop in the group of the other applications or desktops.

7. The method of claim 1, wherein the first or second applications or desktops are the first two applications or desktops within the window stack.

8. The method of claim 1, wherein an order of the first or second applications or desktops within the window stack can be changed by executing a user input gesture, comprising at least one of a flick or drag gesture, on the first and second displays.

9. The method of claim 5, wherein an order of the other applications or desktops within the window stack can be changed by executing a user input gesture, comprising at least one of a flick or drag gesture executed on a selected one of the other applications or desktops within the management window to thereby move the selected one to another position within the window stack.

10. The method of claim 9, wherein a flick or drag gesture executed in a left direction causes the selected one to move down in the stack a predetermined number of positions, and a flick or drag gesture executed in a right direction causes the selected one to move up in the stack a predetermined number of positions.

11. The method of claim 5, wherein separate management windows appear on said first and second displays denoting a first window stack for the first display and a second window stack for the second display, and wherein the other applications or desktops are arranged linearly within each management window from left to right, and the left to right order of the other applications or desktops correspond to an order of the other applications or desktops within the window stack for each corresponding display.

12. The method of claim 5, wherein a selected one of the other applications or desktops is moved to a new position displayed on the first or second displays by executing another user input gesture on the selected one, the another user input gesture comprising at least one of a tap or long hold gesture.

13. A non-transitory computer-readable medium having stored thereon instructions that cause a computing system to execute a method, the instructions comprising:

instructions configured to receive a first predetermined input that represents an instruction to determine and reveal a first desktop or application on a first display of the multi-screen device;

instructions configured to receive a second predetermined input that represents an instruction to determine and reveal a second desktop or application on a second display of the multi-screen device;

instructions configured to respond to the first and second predetermined inputs with outputs that cause the first desktop or application to be displayed on the first display and cause the second desktop or application to be displayed on the second display

instructions configured to receive a third predetermined input that represents an instruction to launch an application manager feature;

instructions configured to respond to the third predetermined input that causes a display of a management window showing all other applications or desktops currently running;

instructions configured to receive a fourth predetermined input that represents an instruction to selectively move one of said first and second applications or desktops, or said other applications or desktops, within the window stack to a different position;

instructions configured to respond to the fourth predetermined input that causes displaying of the window stack in a new order.

14. The computer-readable medium of claim 13, wherein said fourth predetermined input step is executed a number of times by a user until the window stack is displayed in the desired order.

15. The computer-readable medium of claim 13 wherein said management window appears on said first and second displays, and the other applications or desktops are arranged linearly within the management window from left to right, and the left to right order of the other applications or desktops corresponds to an order of the other applications or desktops within the window stack.

16. The computer-readable medium of claim 13, wherein the first or second applications or desktops are the first two applications or desktops within the window stack, and an order of the first or second applications or desktops within the window stack can be changed by executing a user input gesture, comprising at least one of a flick or drag gesture, on the first and second displays.

17. A multi-screen user device, comprising:

a first display including a first display area;

a second display including a second display area;

a first user input gesture area of the first display;

a second user input gesture area of the second display, wherein the first and second user input gesture areas are configured to accept input ftom a user.

a computer-readable medium having instructions stored thereon that include:

a first set of instructions configured to determine a number and identity of desktops or applications selected to be run by a user;

a second set of instructions configured to determine, based on the number and identity of the desktops or applications running, a window stack comprising a logical representation of the desktops and applications within an ordered group whereby a user can selectively move a selected desktop or application to either a visible position on another display or a non-visible position; and

a third set of instructions responsive to a gesture made by the user that represents an instruction to launch an application manager feature, wherein the applications or desktops are displayed on the first and second displays, and a management window is displayed showing all other applications or desktops currently running;

a fourth set of instructions responsive to one or more gestures made by the user that represent an instruction to selectively move one of said first and second applications or desktops, or said other applications or desktops, within the window stack to a different position, and

wherein the window stack is displayed in a new order in response to said fourth set of instructions.

18. The device of claim 17, wherein said fourth set of instructions are executed a number of times until the window stack is displayed in the desired order.

19. The device of claim 17, wherein at least one management window appears on said first and second displays, and the other applications or desktops are arranged linearly within the management window ftom left to right, and the left to right order of the other applications or desktops corresponds to an order of the other applications or desktops within the window stack

20. The device of claim 19, wherein separate management windows appear on said first and second displays denoting a first window stack for the first display and a second window stack for the second display, and wherein the other applications or desktops are arranged linearly within each management window from left to right, and the left to right order of the other applications or desktops correspond to an order of the other applications or desktops within the window stack for each corresponding display.

ABSTRACT

A dual-screen user device and methods for revealing a combination of desktops on single and multiple screens are disclosed. Selected desktops and/or running applications are displayed on dual screen displays. Desktops and applications can be shifted between screens by user gestures, and/or moved off of the screens and therefore hidden. Hidden desktops and screens can be re-displayed by yet other gestures. The desktops and applications are arranged in a window stack that represents a logical order of the desktops and applications providing a user with an intuitive ability to manage multiple applications/desktops running simultaneously. One user gesture launches an applications management window that provides visual indications of all of the applications and desktops running at the time, applications/desktops displayed on the screens. Other gestures can rearrange the order of all of the applications and desktops in the window stack.

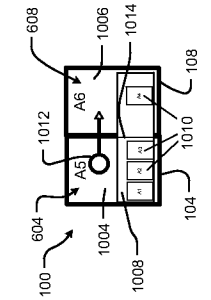


FIG. 10B

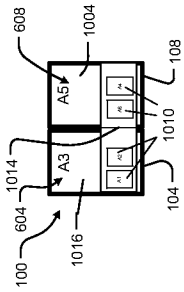


FIG. 10D

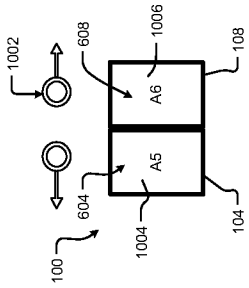


FIG. 10A

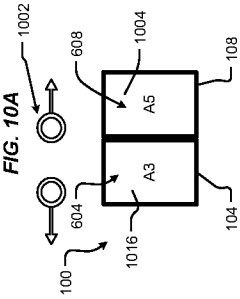


FIG. 10C

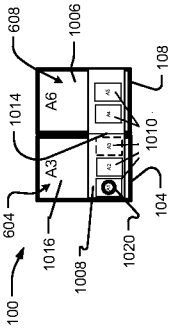


FIG. 10F

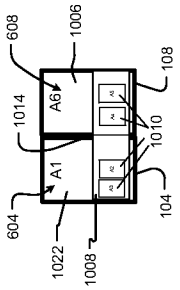


FIG. 10H

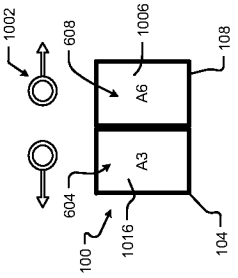


FIG. 10E

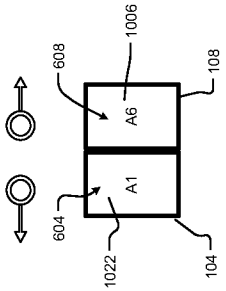


FIG. 10G

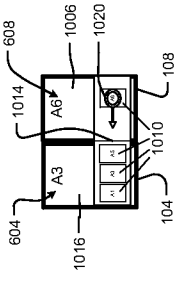


FIG. 10J

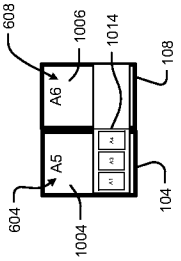


FIG. 10M

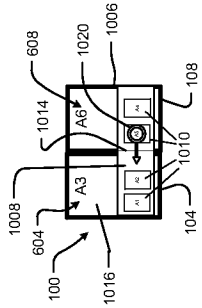


FIG. 10I

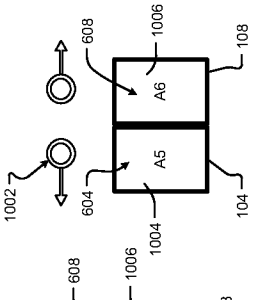


FIG. 10L

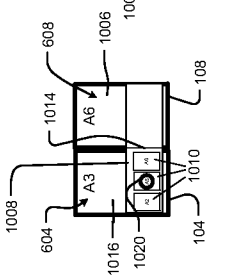


FIG. 10K

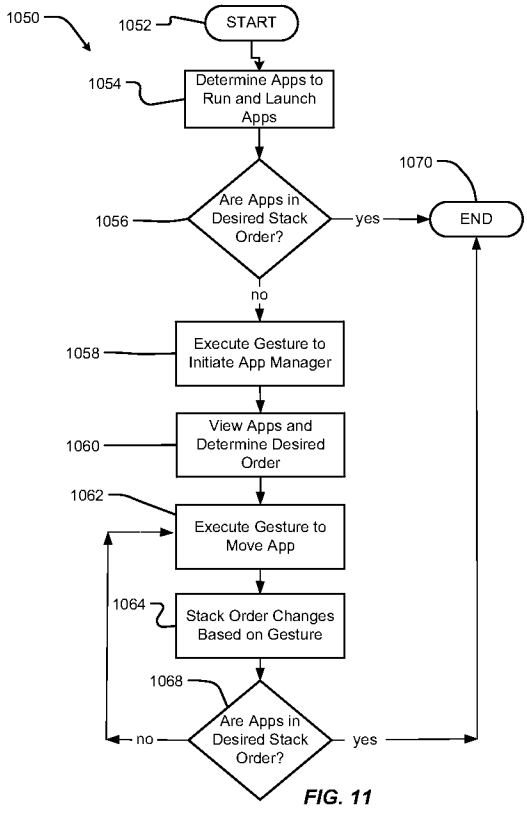


FIG. 11

Brief Description of the Drawings

[0001] FIG. 7 is a first representation of a preferred embodiment showing a device incorporating an annunciator window or display, and an enlarged exploded view of the annunciator window disposed above the device; and

[0002] FIG. 8 is another representation of the embodiment shown in FIG. 7 illustrating the annunciator window displaying an alarm or message.

Detailed Description

[0003] Referring to Fig. 7, the device 100 is depicted with the primary screen 104 and the secondary screen 108 displaying desk tops D1 and D2 on pages 1120 and 1122, respectively. Fig. 7 also illustrates the device in which both screens are in the portrait mode 604 and 608. An annunciator display or window 1100 is shown as extending continuously across both of the screens 104 and 108. The annunciator window is further illustrated as being a unitary display window or display, as opposed to being a separate display on each screen. This annunciator window 1100 can serve many purposes to include the display of messages, alerts, statuses of the device, and other information. In an embodiment, the annunciator display extends across the display areas of said first and second screens for displaying at least one of a device status, a connectivity status, and a messaging status;

[0004] The annunciator window 1100 is provided in the expanded dual screen configuration in order to better view information provided within the window 1100. For annunciator notifications, these may include icons that indicate an event has taken place with respect to a particular function of the device. The Fig. 7 graphically displays boxes 1102, 1104, 1106 and 1107 which can be reserved for receiving and displaying notifications such as incoming phone calls, emails, text messages, and calendar items, respectively. For example, for an incoming call, the phone icon 1102 could flash, change color or otherwise change its visual appearance to signify that an incoming call has been received. One preferred arrangement for the annunciator window 1100 is to align the notifications or messaging statuses over the primary screen 104 as shown. Fig. 7 illustrates other example annunciations displayed in the annunciation window 1100.

[0005] These include, for example, an icon 1108 representing the type of wireless data being exchanged at the time, (such as 3G, 4G, etc.), icon 1110 representing the battery power level, and icon 1112 representing a day and time indicator. Because of the space provided in the annunciator window 1100, a number of other notifications can be represented with appropriate icons in a manner that are clearly visible for the user.

[0006] Now referring to Fig. 8, the device 100 of Fig. 7 is shown when it displays an example notification 1124, such as a notification for a calendar item 1107 having a fairly lengthy text content displayed directly within the window 1100. Other of the annunciator icons are still displayed, such as the wireless data icon 1108, power level 110 and date/time indicator 1112. In another aspect of the invention, it is also contemplated that a longer alert or message having greater text or data content can span across the entire annunciator window 1100, thereby temporarily masking or covering the other icons for a period of time until the alert or message is cleared by the user.

[0007] Therefore, it is apparent from the Figs. 7 and 8 that alerts, messages, statuses, and other annunciations are capable of spanning across both screens simultaneously, thereby providing more information to the user. Because of the increased area dedicated for the various annunciations, this helps ensure the user is better able to view the annunciations, thereby making the device more user friendly and efficient. The annunciated window can be provided without materially degrading the presentation of displayed information on the primary and secondary screens, since the annunciator window can be placed near the top or bottom edges of the device where typically critical data/images are not displayed.

Claims

[0008] 1. A method of displaying information on a multi-screen device including a plurality of desktops and/or applications each having at least one window, and an annunciator window, the method comprising:

receiving, by a processor, a first input that represents an instruction to reveal one of a desktop or application on a first display of the multi-screen device and selecting a first desktop or application to display on the first display;

receiving, by a processor, a second input that represents an instruction to reveal one of a desktop or application on a second display of the multi-screen device and selecting a second desktop or application to display on the second display;

displaying, by a processor, the selected first and second desktops or applications on the first and second displays;

displaying, by a processor, a pre-configured annunciator window having information therein showing at least one of a device status, a connectivity status, and a messaging status;

configuring by a processor, the annunciator window to extend across both said first and second displays;

modifying, by a processor, the display of the annunciator window in response to a changed state of at least one of said device status, a connectivity status, and messaging status; and

wherein the modified display of the annunciator window displays information selectively across substantially all or part of said annunciator window.

[0009] 2. The method of claim 1, wherein said modifying step includes displaying the changed state including text corresponding to at least one of a phone message, email message, or text message.

[0010] 3. The method of claim 1, wherein said modifying step includes displaying the changed state across a portion of said annunciator window and masking at least one icon representing said device status, a connectivity status, or messaging status.

[0011] 4. The method of claim 1, wherein said modifying step includes displaying the changed state across substantially all of said annunciator window.

[0012] 5. The method of claim 1, wherein said annunciator window is located across adjacent upper or lower edges of said first and second displays.

[0013] 6. A non-transitory computer-readable medium having stored thereon instructions that cause a computing system to execute a method, the instructions comprising:

instructions configured to receive a first predetermined input that represents an instruction to determine and reveal a first desktop or application on a first display of the multi-screen device;

instructions configured to receive a second predetermined input that represents an instruction to determine and reveal a second desktop or application on a second display of the multi-screen device;

instructions configured to respond to the first and second predetermined inputs with outputs that cause the first desktop or application to be displayed on the first display and cause the second desktop or application to be displayed on the second display

instructions configured to receive a third predetermined input that represents an instruction to display a pre-configured annunciator window having information therein showing at least one of a device status, a connectivity status, and a messaging status; and

instructions configured to respond to the third predetermined input with an output that causes the annunciator window to be displayed in a configuration extending across both of said first and second displays.

[0014] 7. The computer-readable medium of claim 6, further including:

instructions configured to receive a fourth input that represents instructions to display a changed state of said device including text corresponding to at least one of a phone message, email message, or text message; and

instructions configured to respond to the fourth input with an output that causes the annunciator window to be displayed in a changed configuration extending across both of said first and second displays.

[0015] 8. The computer-readable medium of claim 7, wherein said instructions configured to respond to the fourth input cause a changed configuration to be displayed masking at least one icon representing said device status, a connectivity status, or messaging status.

[0016] 9/ The computer-readable medium of claim 7, wherein said instructions configured to respond to the third predetermined input causes said annunciator window to be displayed across adjacent upper or lower edges of said first and second displays.

[0017] 10. A multi-screen user device, comprising:

a first display including a first display area;

a second display including a second display area;

a first user input gesture area of the first display;

a second user input gesture area of the second display, wherein the first and second user input gesture areas are configured to accept input from a user;

an annunciator display extending across said first and second display areas of said first and second displays for displaying at least one of a device status, a connectivity status, and a messaging status;

a computer-readable medium having instructions stored thereon that include:

a first set of instructions configured to determine a number and identity of desktops or applications selected to be run by a user; and

a second set of instructions configured to determine information to be displayed in said annunciator display based on selected user gestures and respective states of said device status, a connectivity status, and a messaging status;

[0018] 11. The device of claim 10, wherein: the second set of instructions include information regarding a changed state of one of said statuses including text corresponding to at least one of a phone message, email message, or text message.

[0019] 12. The device of claim 10, wherein said second set of instructions result in displaying a changed state of said annunciator window across a portion of said annunciator window and masking at least one icon representing said device status, a connectivity status, or messaging status.

[0020] 13. The device of claim 10, wherein said second set of instructions result in displaying a changed state of said annunciator window across substantially all of said annunciator window.

[0021] 14. The device of claim 10, wherein said second set of instructions result in displaying said annunciator window across adjacent upper or lower edges of said first and second displays.

ABSTRACT

A dual-screen user device and methods for revealing a combination of desktops on single and multiple screens are disclosed. Selected desktops and/or running applications are displayed on dual screen displays. Desktops and applications can be shifted between screens by user gestures, and/or moved off of the screens and therefore hidden. Hidden desktops and screens can be re-displayed by yet other gestures. The desktops and applications are arranged in a window stack that represents a logical order of the desktops and applications providing a user with an intuitive ability to manage multiple applications/desktops running simultaneously. User gestures can rearrange the order of all of the applications and desktops in the window stack. One embodiment provides an annunciator display or window extending across both screens in a dual screen configuration of the device. The annunciator window provides alerts, notifications, and statuses of the device in an increased area thereby enhancing viewability of the information in the window.

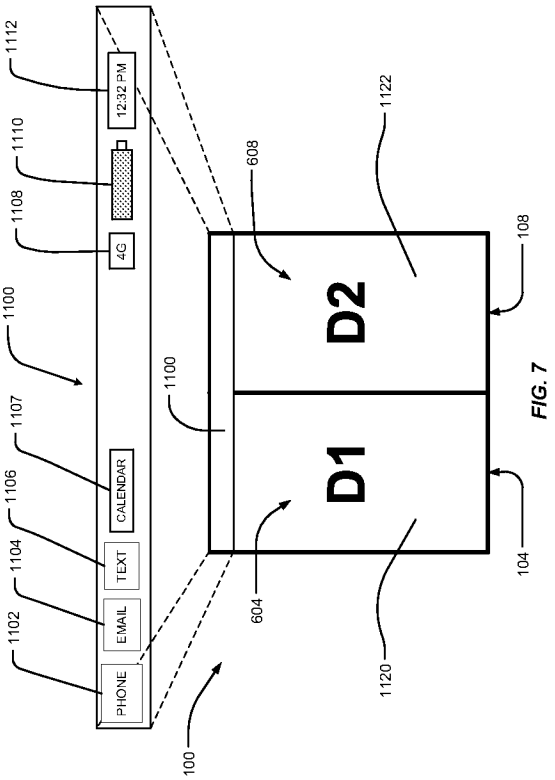

1112
12:32 PM
1110
1108
4G
1100
1107
CALENDAR
1106
TEXT
1104
EMAIL
1102
PHONE
1100
608
1122
108
D2
D1
604
1120
104
100

FIG. 7

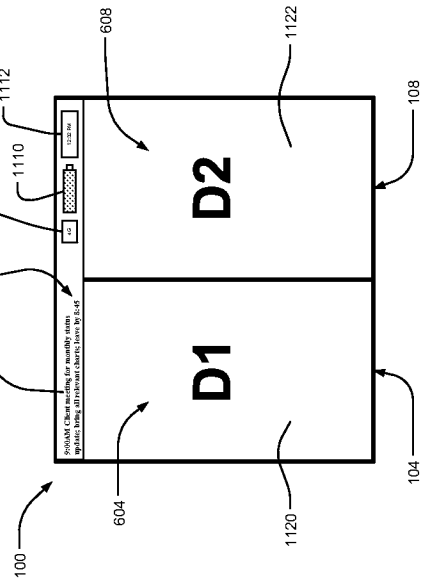

608
1122
1112
1110
1108
1124
1100
D2
D1
108
604
1120
104
100

FIG. 8

[0001] Brief Description of the Drawings

[0002] FIG. 7 is a first representation of a preferred embodiment showing a device incorporating an annunciator window or display, and an enlarged exploded view of the annunciator window disposed above the device; and

[0003] FIG. 8 is another representation of the embodiment shown in FIG. 7 illustrating the annunciator window displaying an alarm or message.

[0004] FIG. 9A is a first representation of another preferred embodiment showing the device incorporating an annunciator window or display;

[0005] FIG. 9B is a second representation of the preferred embodiment of FIG 9A showing the device incorporating an annunciator window or display and a user executing a drag gesture on the primary screen to open the annunciator window drawer to provide full access to the contents of the annunciator window;

[0006] FIG. 9C is a representation of the preferred embodiment of FIG 9B showing the device with the annunciator window drawer opened or expanded on the primary screen;

[0007] FIG. 9D is a representation of the preferred embodiment of FIG 9C showing the device with the annunciator window drawer opened or expanded, and the user executing another drag gesture to close or minimize the annunciator window drawer;

[0008] FIG. 9E is another representation of the preferred embodiment of FIG 9A showing the device incorporating the annunciator window or display and a user executing a drag gesture on the secondary screen to open the annunciator window drawer;

[0009] FIG. 9F is a representation of the preferred embodiment of FIG 9E showing the device with the annunciator window drawer opened or expanded on the secondary screen; and

[0010] FIG. 9G is a representation of the preferred embodiment of FIG 9F showing the device with the annunciator window drawer opened or expanded, and the user executing another drag gesture to close or minimize the annunciator window drawer.

Detailed Description

[0011] Referring to Fig. 7, the device 100 is depicted with the primary screen 104 and the secondary screen 108 displaying desk tops D1 and D2 on pages 1120 and 1122, respectively. Fig. 7 also illustrates the device in which both screens are in the portrait mode 604 and 608. An annunciator display or window 1100 is shown as extending continuously across both of the screens 104 and 108. The annunciator window is further illustrated as being a unitary display window or display, as opposed to being a separate display on each screen. This annunciator window 1100 can serve many purposes to include the display of messages, alerts, statuses of the device, and other information. In an embodiment, the annunciator display extends across the display areas of said first and second screens for displaying at least one of a device status, a connectivity status, and a messaging status;

[0012] The annunciator window 1100 is provided in the expanded dual screen configuration in order to better view information provided within the window 1100. For annunciator notifications, these may include icons that indicate an event has taken place with respect to a particular function of the device. The Fig. 7 graphically displays boxes 1102, 1104, 1106 and 1107 which can be reserved for receiving and displaying notifications such as incoming phone calls, emails, text messages, and calendar items, respectively. For example, for an incoming call, the phone icon 1102 could flash, change color or otherwise change its visual appearance to signify that an incoming call has been received. One preferred arrangement for the annunciator window 1100 is to align the notifications or messaging statuses over the primary screen 104 as shown.

[0013] Fig. 7 illustrates other example annunciations displayed in the annunciation window 1100. These include, for example, an icon 1108 representing the type of wireless data being exchanged at the time, (such as 3G, 4G, etc.), icon 1110 representing the battery power level, and icon 1112 representing a day and time indicator. Because of the space provided in the annunciator window 1100, a number of other notifications can be represented with appropriate icons in a manner that are clearly visible for the user.

[0014] Now referring to Fig. 8, the device 100 of Fig. 7 is shown when it displays an example notification 1124, such as a notification for a calendar item 1107 having a fairly lengthy text content displayed directly within the window 1100. Other of the annunciator icons are still displayed, such as the wireless data icon 1108, power level 110 and date/time indicator 1112. In another aspect of the invention, it is also contemplated that a longer alert or message having greater text or data content can span across the entire annunciator window 1100, thereby temporarily masking or covering the other icons for a period of time until the alert or message is cleared by the user.

[0015] Therefore, it is apparent from the Figs. 7 and 8 that alerts, messages, statuses, and other annunciations are capable of spanning across both screens simultaneously, thereby providing more information to the user. Because of the increased area dedicated for the various annunciations, this helps ensure the user is better able to view the annunciations, thereby making the device more user friendly and efficient. The annunciated window can be provided without materially degrading the presentation of displayed information on the primary and secondary screens, since the annunciator window can be placed near the top or bottom edges of the device where typically critical data/images are not displayed.

[0016] Now referring to Fig. 9A, the device 100 is illustrated again with the annunciator window 1100 that spans across both the primary and secondary screens 104 and 108. In this example, application A1 is running on page 1004, while an application A2 is running on page 1006.

[0017] Referring to Fig. 9B, this figure illustrates a user executing a drag gesture 1002 from the annunciator window 1100, and the user has chosen to execute the drag gesture over the primary screen 104 in order to open the annunciator window "drawer", that is to expand the annunciator window over the primary screen 104.

[0018] Now referring to Fig. 9C, the annunciator window 1100 has now been expanded over the primary screen 104, and now appears as an expanded annunciator window 1105 which covers most of the primary screen 104. By executing this gesture, the user is able to see all of the information within the annunciator window which is aligned over the primary screen. Referring back to the example of Fig. 7 and 8, the user could therefore view all text or images associated with any alerts, messages, or other annunciator information that could be corresponding to phone calls, emails, text messages or other information.

[0019] Referring to Fig. 9D, if the user wishes to close the annunciator drawer 1105, the user can simply execute another drag gesture 1002 to return the annunciator window 1100 to its original configuration. Alternatively, another gesture could be used such as a tap gesture to minimize or close the drawer.

[0020] Referring now to Fig. 9E, this figure illustrates the user executing the drag gesture 1002 over the secondary screen 108. Accordingly, as shown in Fig. 9F, the annunciator drawer 1105 is illustrated as an expansion of the annunciator window 1100 over the secondary screen 108. Once again, the user may now view the full information that may be appear on the annunciator window 1100, such as various connectivity information, battery status, time, date, etc.

[0021] Referring to Fig. 9G, the user has again executed another gesture 1002 in order to close the annunciator window 1105, thereby returning the annunciator window to its original configuration.

[0022] From the foregoing, it is apparent that the capability of expanding the size of the annunciator window provides a number of advantages to the user. Once clear advantage is that a user does not have to close out or minimize applications that may be running at the time on applications or desktops displayed on the screens and rather, the user may simply execute a gesture in order to expand or contract the annunciator window size.

Claims

1. A method of displaying information on a multi-screen device including a plurality of desktops and/or applications each having at least one window, and an annunciator window, the method comprising:

receiving, by a processor, a first input that represents an instruction to reveal one of a desktop or application on a first or second display of the multi-screen device and selecting a desktop or application to display on the first or second display;

displaying, by a processor, the selected desktop or application on the first or second displays;

displaying, by a processor, a pre-configured annunciator window having information therein showing at least one of a device status, a connectivity status, and a messaging status;

configuring, by a processor, the annunciator window to extend across both said first and second displays;

expanding, by a processor, a size of the annunciator window in response to a user input gesture wherein the annunciator is expanded as a drawer over a selected one of said first or second displays; and

wherein the annunciator window displays information selectively across substantially all or part of said expanded annunciator window.

2. The method of claim 1, further including:

modifying, by a processor, the display of the annunciator window in response to a changed state of at least one of said device status, a connectivity status, and messaging status.

3. The method of claim 2, wherein said modifying step includes displaying the changed state including text corresponding to at least one of a phone message, email message, or text message.

4. The method of claim 2, wherein said modifying step includes displaying the changed state across a portion of said annunciator window and masking at least one icon representing said device status, a connectivity status, or messaging status.

5. The method of claim 1, wherein said modifying step includes displaying the changed state across substantially all of said annunciator window.

6. The method of claim 1, further including:

positioning, by a processor, the annunciator window across adjacent upper or lower edges of said first and second displays.

7. The method of claim 1, wherein the annunciator window is expanded across a selected one of said first or second displays in response to the user input gesture executed on the selected display.

8. The method of claim 1, wherein the selected desktop or application runs on the first or second display, the user executes the user input gesture on the same display upon which the desktop or application is running resulting in the annunciator window expanding over the desktop or application without the user having to close or minimize the desktop or application.

9. The method of claim 8, wherein the user executes another user input gesture to minimize or close the annunciator window drawer causing the running desktop or application be re-displayed on the same first or second display.

10. A non-transitory computer-readable medium having stored thereon instructions that cause a computing system to execute a method, the instructions comprising:

instructions configured to receive a first input that represents an instruction to determine and reveal a desktop or application on a first or second display of the multi-screen device;

instructions configured to respond to the first input with an output that cause the desktop or application to be displayed on the first or second displays;

instructions configured to receive a second input that represents an instruction to display a pre-configured annunciator window having information therein showing at least one of a device status, a connectivity status, and a messaging status;

instructions configured to respond to the second input with an output that causes the annunciator window to be displayed in a configuration extending across both of said first and second displays;

instructions configured to receive a third input that represents an instruction to expand the annunciator window over a selected one of said first or second displays;

instructions configured to respond to the third input with an output that causes an annunciator window drawer to be expanded over the selected one of said first or second displays.

11. The computer-readable medium of claim 10, further including:

instructions configured to receive a fourth input that represents instructions to display a changed state of said device including a display of text corresponding to at least one of a phone message, email message, or text message; and

instructions configured to respond to the fourth input with an output that causes the annunciator window to be displayed in a changed configuration extending across both of said first and second displays.

12. The computer-readable medium of claim 10, wherein the third input is a user gesture executed on the same display upon which the desktop or application is running, resulting in the annunciator window expanding over the desktop or application without the user having to close or minimize the desktop or application.

13. The computer-readable medium of claim 10, further including:

instructions configured to receive a fourth input that represents an instruction to minimize or close the annunciator window on the selected one of said first or second displays; and

instructions configured to respond to the fourth input with an output that causes the annunciator window drawer to be minimized or closed causing the running desktop or application be re-displayed on the same first or second display.

14. The computer-readable medium of claim 11, wherein said instructions configured to respond to the fourth input cause a changed configuration to be displayed masking at least one icon representing said device status, a connectivity status, or messaging status.

15. The computer-readable medium of claim 11, wherein said instructions configured to respond to the second predetermined input causes said annunciator window to be displayed across adjacent upper or lower edges of said first and second displays.

16. A multi-screen user device, comprising:

a first display including a first display area;

a second display including a second display area;

a first user input gesture area of the first display;

a second user input gesture area of the second display, wherein the first and second user input gesture areas are configured to accept input from a user;

an annunciator display extending across a portion said first and second display areas of said first and second displays for displaying at least one of a device status, a connectivity status, and a messaging status;

a computer-readable medium having instructions stored thereon that include:

a first set of instructions configured to determine a number and identity of desktops or applications selected to be run by a user and subsequently displayed on selected ones of said first or second displays;

a second set of instructions configured to determine information to be displayed in said annunciator display based on selected user gestures and respective states of said device status, a connectivity status, and a messaging status; and

a third set of instructions configured to expand an area of the annunciator display defining a drawer, as presented on the first or second displays in response to a user gesture executed on a selected display;

17. The device of claim 16, wherein: the second set of instructions include information regarding a changed state of one of said statuses including text displayed corresponding to at least one of a phone message, email message, or text message.

18. The device of claim 16, wherein said second set of instructions result in displaying said annunciator window across adjacent upper or lower edges of said first and second displays.

19. The device of Claim 16, wherein the third set of instructions result in the annunciator display expanding over a corresponding desktop or application without further instructions required to close or minimize displayed desktops or applications.

20. The device of Claim 16, further including:

a fourth set of instructions configured to minimize or close the annunciator window drawer on the first or second displays in response to another user gesture executed on the selected display causing the desktop or application to be re-displayed on the same first or second displays.

ABSTRACT

A dual-screen user device and methods for revealing a combination of selected desktops and applications on single and dual screens are disclosed. Desktops and applications can be shifted between screens by user gestures, and/or moved off of the screens and therefore hidden. Hidden desktops and screens can be re-displayed by other gestures. The desktops and applications are arranged in a window stack that represents a logical order of the desktops and applications providing a user with an intuitive ability to manage multiple applications/desktops running simultaneously. One embodiment provides an annunciator window extending across both screens in a dual screen configuration. The annunciator window provides alerts, notifications, and statuses of the device in an increased area thereby enhancing viewability of the information in the window. The annunciator window can be expanded over a selected screen to view full contents of the window without having to minimize or close running applications.

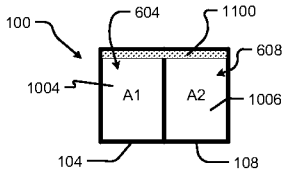


FIG. 9A

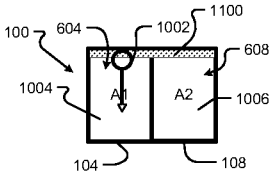


FIG. 9B

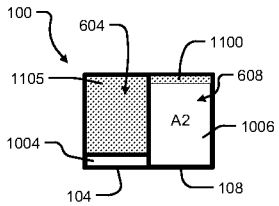


FIG. 9C

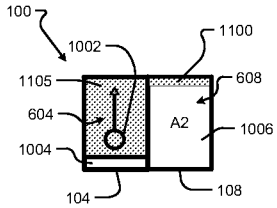


FIG. 9D

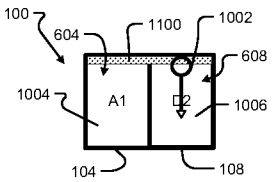


FIG. 9E

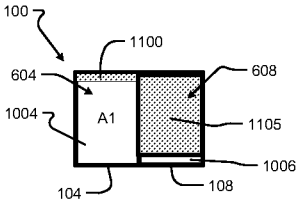


FIG. 9F

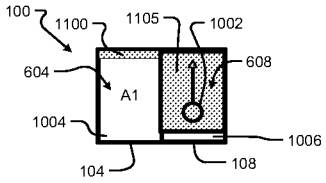


FIG. 9G

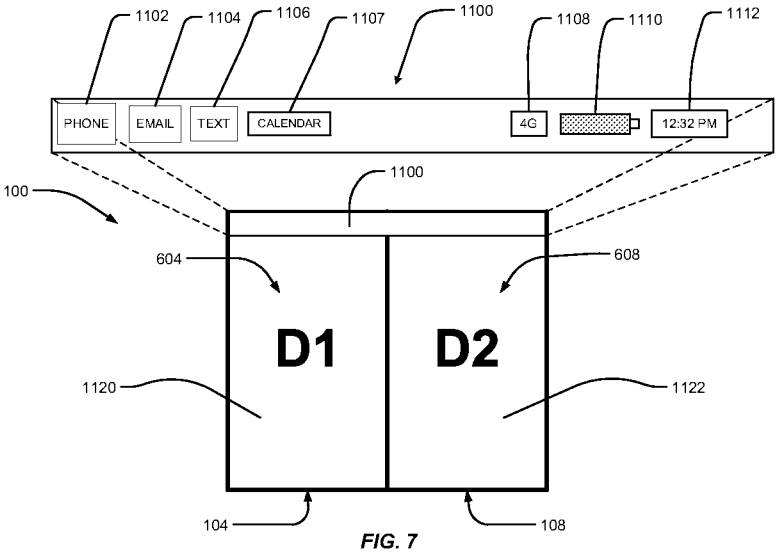


FIG. 7

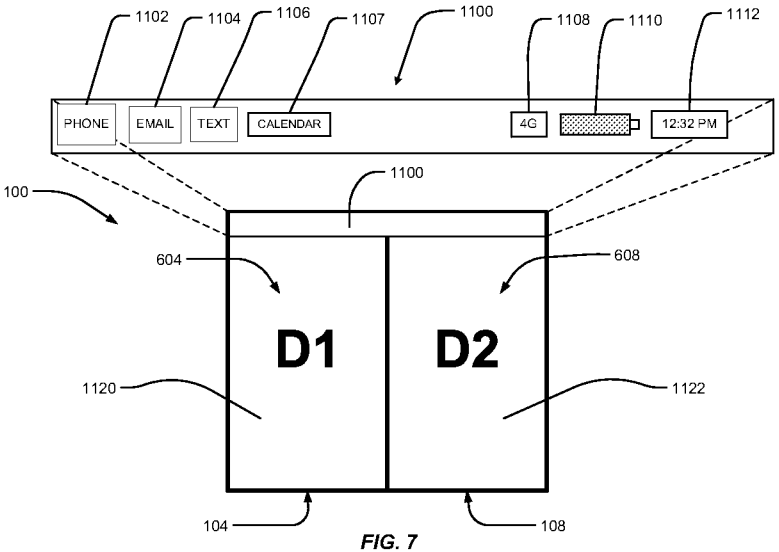


FIG. 7

[0001] Brief Description of the Drawings

[0002]

[0003] FIG. 10A depicts a first display state of an open device with a visual indicator on the primary screen indicating the application displayed on the primary screen can be maximized to also be shown in a larger view across both the primary and secondary screens,

[0004] FIG. 10B depicts a second display state of the open device of FIG. 10A showing a user gesture executed to maximize the application, and a graphical representation of the selected application expanding to occupy both the primary and secondary screens in response to the user gesture;

[0005] FIG. 10C depicts a third display state of the open device of FIG. 10A showing the selected application maximized and therefore displayed on both the primary and secondary screens;

[0006] FIG. 10D depicts a fourth display state of the open device of FIG. 10A showing the selected application maximized as in FIG. 10C, but also another user gesture executed to minimize the application on the secondary screen; and

[0007] FIG. 10E depicts a fifth display state of the open device of FIG. 10A showing the selected application minimized on the secondary screen thereby covering the previous application appearing on the secondary screen , and also revealing the next application or desktop in a window stack on the primary screen.

Detailed Description

[0008] Figs. 10A-E depict graphical representations of embodiments of a device 100 in an open portrait state where the primary screen or display 104 in a first portrait configuration 604 is shown surrounded by a solid line, while the secondary screen or display 108 is in a second portrait configuration 608 and is also visible as indicated by a solid line.

[0009] Referring to Fig. 10A, the primary screen 104 shows a first page 1004 running a first single page application A1, and the secondary display 108 shows another page 1006 running a second desk top D2. Fig. 10A also illustrates a visual indicator 1002 indicating to the user that the application or a desk top running on the screen in which the visual indicator is displayed allows the user to minimize or maximize that application/desktop. Accordingly, in Fig. 10A, the indicator 1002 signifies that a user can maximize the application A1 in its current minimized state. The indicator 1002 is shown as having a wave shape that extends across the upper portion of the screen 104; however, it shall be understood that other graphical representations can be provided for the indicator 1002 such as an icon or another type of visual indicator located on other parts of the screen. The visual indicator 1002 can be presented within an annunciator window 1003, or can be presented without a separate window and directly on the running application/desktop, preferably in the manner shown at the top of the screens so as to not interfere substantially with the applications/desktops. The annunciator window 1003 can also display other features of the device to include notifications (such as incoming calls, emails, or text messages)(not shown). The annunciator window 1003 could also display status information such as whether the device has a 3G or 4G connection, the date, time, battery charge level, wireless signal strength, and other status information (not shown).

[0010] Referring to Fig. 10B, this figure illustrates a user executing a gesture 1008, such as a tap or long hold gesture on the visual indicator 1002. In response to the gesture 1008, the application A1 is maximized for viewing on both the first and second screens. Fig. 10B illustrates the application A1 in a moving state as it expands onto the secondary screen 108. Although Fig. 10B may illustrate a gradual or incremental maximizing of the application A1, it shall be understood that Fig. 10B may also represent that once the use executes the gesture 1008, the selected application/desk top will be instantaneously shown on both of the screens.

[0011] Referring to Fig. 10C, this Figure shows the application A1 as it now fully occupies both the primary and secondary screens. The visual indicator 1002 therefore also expands across both the primary and secondary screens, indicating that the user can now minimize this application by again executing another gesture.

[0012] Referring to Fig. 10D, this figure illustrates the user executing another gesture 1008 on the secondary screen 108, which therefore minimizes the application A1 onto the secondary screen 108. Again, Fig. 10D illustrates a gradual or incremental movement of the application A1, however, it shall again be understood that Fig. 10D is only a graphical representation of the minimization, and thus the minimization could occur nearly simultaneously as viewed by the user.

[0013] Referring to Fig. 10E, in yet another embodiment of the invention, the application A1 is shown as minimized on the secondary screen 108, while the primary screen 104 now shows a new page 1010 running an application A2. In this example, the minimization of the application A1 results in rearranging the window stacks for the primary and secondary screens. Therefore, by the user executing the gesture 1008 on the secondary screen to minimize application A1, a next application/desktop in the window stack for the primary screen 104 is displayed, namely the application A2. The application A1 is now also moved to the top application in a window stack of the secondary screen, assuming the secondary screen has its own window stack. Alternatively, if there is but a single window stack for both screens, then the page 1004 running the application A1 appearing on the secondary screen 108 can also represent rearrangement of the single window stack in which page 1004 is now the first or top application to occupy the secondary screen and the page 1010 running the application A2 is now the first or top application occupying the primary screen. The page 1006 running the application D2 is placed one level below the application A1 in the window stack, or could be placed at any other location in the window stack depending upon the particular window stack protocol being employed at the time. Yet further in the alternative, the minimization of the application A1 onto the secondary screen could result in no rearrangement of any window stack, and the primary screen 104 could be blank with no displayed application/desktop, or the primary screen could display a predetermine default application/desktop.

[0014] If the user desired to maximize application A1 again, the user could simply execute yet another gesture 1008 on the secondary screen 108, which would cause the application A1 to be displayed on both the primary and secondary screens such as shown in Fig. 10C.

[0015] Although Figs. 10A-10E are presented in the open portrait state, it shall be understood that the features disclosed in these figures can also be incorporated in other modes of operation of the device 100 to include a landscape state or any of the other "physical" states disclosed herein.

Based on the description of the Figures, it is evident that the display of windows in a window stack can be selectively changed by simple gesture interactions thereby making multiple desktops and running applications available to the user with minimum effort. Each screen has a logical representation of the windows that have been opened and displayed on the screens. The logical representation is the stack that is made up of one or more windows arranged in an order that can be selected and changed by the user. The windows can be indexed or shifted to display desire windows, as well as to move windows between separate stacks. The ordered arrangement of a stack maintains the logical construction of the stack intact when the user navigates the window stack, for example in an off-screen area. As shown specifically in the Figures 10A-10E, a user may easily manipulate the display of selected running applications or desktops by minimizing and maximizing them. The user may also reveal at least the next desktop or application in a stack if the user moves an application/desktop to another screen.

Claims

1. A method of displaying a plurality of desktops and/or applications, each having at least one window, on a multi-screen device, comprising:

receiving, by a processor, a first input that represents an instruction to reveal one of a desktop or application on a first display of the multi-screen device;

selecting, by a processor, a first desktop or application to display on the first display;

receiving, by a processor, a second input that represents an instruction to reveal one of a desktop or application on a second display of the multi-screen device;

selecting, by a processor, a second desktop or application to display on the second display;

displaying, by a processor, the selected first and second desktops or applications on the first and second displays;

receiving, by a processor, a third input that represents an instruction to maximize or minimize a selected one of the first or second desktops or applications;

displaying, by a processor, the selected one of the first or second desktops or applications in a maximized or minimized condition

2. The method of claim 1, wherein at least one of the first, second, and third inputs is achieved by at least a user input gesture.

3. The method of claim 1, wherein the third input is a user gesture comprising at least one of a tap or long hold gesture.

4. The method of claim 1, wherein at least one of the first or second displays has a visual indicator displayed thereon and signifying that a desktop or application appearing on that display can be maximized or minimized by a user gesture.

5. The method of claim 1, wherein the visual indicator has a wave shape.

6. The method of claim 1, wherein the third input is an instruction to maximize a selected one of the first or second desktops or applications, resulting in the selected one being displayed on both of the first and second displays.

7. The method of claim 1, wherein the third input is an instruction to minimize a selected one of the first or second desktops or applications, resulting in the selected one being displayed only on either the first and second displays.

8. A non-transitory computer-readable medium having stored thereon instructions that cause a computing system to execute a method, the instructions comprising:

instructions configured to receive a first predetermined input that represents an instruction to determine and reveal a first desktop or application on a first display of the multi-screen device;

instructions configured to receive a second predetermined input that represents an instruction to determine and reveal a second desktop or application on a second display of the multi-screen device;

instructions configured to respond to the first and second predetermined inputs with outputs that cause the first desktop or application to be displayed on the first display and cause the second desktop or application to be displayed on the second display;

instructions configured to receive a third predetermined input that represents an instruction to maximize or minimize a selected one of the first or second desktops or applications; and

instructions configured to respond to the third predetermined input that causes displaying of the selected one of the first or second desktops or applications in a maximized or minimized condition.

9. The computer-readable medium of claim 8, wherein at least one of the first or second displays has a visual indicator displayed thereon and signifying that a desktop or application appearing on that display can be maximized or minimized by a user gesture according to the instructions configured to receive a third predetermined input.

10. The computer-readable medium of claim 8 wherein the third predetermined input is an instruction to maximize a selected one of the first or second desktops or applications, resulting in the selected one being displayed on both of the first and second displays according to the instructions configured to respond to the third predetermined input.

11. The computer-readable medium of claim 8 wherein the third predetermined input is an instruction to minimize a selected one of the first or second desktops or applications, resulting in the selected one being displayed only on either the first and second displays according to the instructions configured to respond to the third predetermined input.

12. A multi-screen user device, comprising:

a first display including a first display area;

a second display including a second display area;

a first user input gesture area of the first display;

a second user input gesture area of the second display, wherein the first and second user input gesture areas are configured to accept input from a user;

a computer-readable medium having instructions stored thereon that include:

a first set of instructions configured to determine a number and identity of desktops or applications selected to be run by a user;

a second set of instructions configured to determine, based on the number and identity of the desktops or applications running, a window stack comprising a logical representation of the desktops and applications within an ordered group whereby a user can selectively move a selected desktop or application within the window stack and to be displayed in either the first or second displays;

a third set of instructions responsive to a gesture made by the user to maximize or minimize a selected one of the first or second desktops or applications; and

wherein the selected one of the first or second desktops or applications is displayed in a maximized or minimized condition.

13. The device of claim 12, wherein at least one of the first or second displays has a visual indicator displayed thereon and signifying that a desktop or application appearing on that display can be maximized or minimized by a user gesture.

14. The device of claim 13, wherein the visual indicator has a wave shape.

15. The device of claim 12, wherein the third set of instructions are instructions to maximize a selected one of the first or second desktops or applications, resulting in the selected one being displayed on both of the first and second displays.

16. The device of claim 12, wherein the third set of instructions are instructions to minimize a selected one of the first or second desktops or applications, resulting in the selected one being displayed only on either the first and second displays.

17. The device of clam 13, wherein the visual indicator appears on the applications or desktops running on both the first and second displays so that gestures made by the user to maximize or minimize a selected one of the first or second desktops or applications enables a user to maximize and minimize the applications or desktops appearing on both the first and second displays sequentially and repeatedly.

ABSTRACT

A dual-screen user device and methods for revealing a combination of desktops on single and multiple screens are disclosed. A determined number of desktops and/or running applications can be selectively displayed on dual screen displays conditioned upon inputs received and the state of the device. Desktop displays and applications can be selectively shifted between the screens by user gestures, and can be moved off of the screens by other user gestures and therefore hidden. The hidden desktops and screens however can be re-displayed by yet another gesture. The desktops and applications are arranged in a window stack that represents a logical order of the desktops and applications allowing a user with an intuitive ability to manage multiple applications/desktops running simultaneously. Visual indicators can be used on the displayed applications and desktops enabling a user to maximize a display on multiple screens or to minimize them on a single selected screen.

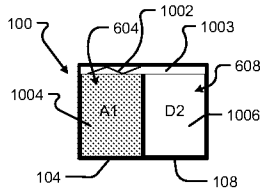


FIG. 10A

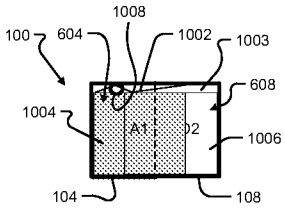


FIG. 10B

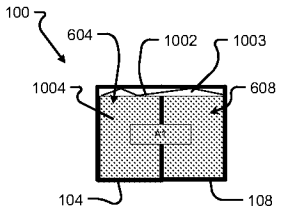


FIG. 10C

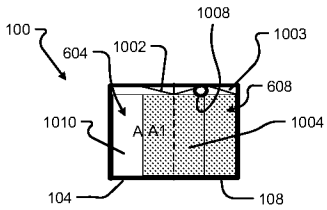


FIG. 10D

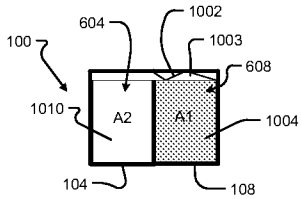


FIG. 10E

Brief Description of the Drawings

[0001] FIG. 10A depicts a first display state of an open device with a depiction of executing a spread gesture to launch an application manager;

[0002] FIG. 10B depicts a second display state of the open device of FIG. 10A with the application manager launched in the form of a separate window or tray that depicts the non-displayed applications and/or desktops running on the device, and also depicting the window stack arrangement of the non-displayed applications and/or desktops; this Figure 10B also displays execution of a user gesture to rearrange the order of the window stack by a gesture executed on the primary screen;

[0003] FIG. 10C depicts a third display state of an open device after execution of the gesture in FIG. 10B , and also another spread gesture to re-launch the application manager;

[0004] FIG. 10D shows a fourth display state of the open device in which the application manager is launched and the non-displayed applications and/or desktops running on the device are shown in the manager window or tray in a new order based on the user gesture executed in FIG. 10B;

[0005] FIG. 10E depicts a first display state of an open device in another embodiment with a depiction of executing a spread gesture to launch an application manager;

[0006] FIG. 10F depicts a second display state of the open device of FIG. 10E with the application manager launched showing the manager window or tray that depicts the non-displayed applications and/or desktops running on the device, and also showing the stack arrangement of the non-displayed applications and/or desktops; this Figure 10F also displays execution of a user gesture to rearrange the order of the window stack by a gesture executed within the application manager to move a non-displayed application or desktop onto the primary screen;

[0007] FIG. 10G depicts a third display state of the open device of FIG. 10E after execution of the gesture in FIG 10F , and also another spread gesture to re-launch the application manager;

[0008] FIG. 10H shows a fourth display state of the open device of FIG 10E in which the application manager is launched and the non-displayed applications and/or desktops running on the device are shown in a new order based on the user gesture executed in FIG. 10F;

[0009] Figure 10I displays a first display state of an open device in yet another embodiment with the application manager launched showing the manager window or tray with multiple applications/desktops, and showing execution of a user gesture in the form of a drag gesture to rearrange the order of the window stack

[0010] FIG 10J displays a second display state of the open device of Figure 10I with the application manager launched showing the manager window or tray with the multiple applications/desktops, and showing execution of another user gesture in the form of a drag gesture to further rearrange the order of the window stack;

[0011] FIG 10K displays a third display state of the open device of Figure 10I with the application manager launched showing the manager window or tray with the multiple applications/desktops, and showing execution of yet another user gesture in the form of a tap gesture to further rearrange the order of the window stack in which the gesture opens the corresponding application/desktop in a predetermined screen;

[0012] FIG 10L displays a fourth display state of the open device of Figure 10I showing the new application displayed in the primary screen and the application manager closed; this Figure further depicts execution of another spread gesture to re-launch the application manager; and

[0013] FIG. 10M displays a fifth display state of the open device of Figure 10I in which the manager window or tray shows the new window stack order responsive to the previous user gestures that rearranged the window stack.

[0014] FIG. 10N depicts a first display state of an open device in another embodiment with a depiction of executing a spread gesture to launch an application manager;

[0015] Figure 10O displays a second display state of an open device in yet another embodiment with the application manager launched showing the manager window or tray with multiple applications/desktops, and showing execution of a user gesture in the form of a drag gesture to rearrange the order of the window stack by dragging an application ftom the manager window to a selected screen;

[0016] FIG. 10P depicts a third display state of the open device

[0017] showing the newly displayed application on one of the screens that was dragged ftom the manager window; this Figure further depicts execution of another spread gesture to re-launch the application manager;

[0018] FIG. 10Q displays a fourth display state of the open device of Figure 10P in which the manager window or tray shows the new window stack order responsive to the previous user gesture that rearranged the window stack; and

[0019] FIG. 11 shows a flow diagram depicting logic for executing rearrangement of a selected window stack.

Detailed Description

[0020] Fig. 10A depicts graphical representations of embodiments of a device 100 in an open portrait state where the primary screen is also visible as indicated by a solid line. The primary screen 104 shows a first page 1004 running a single page application A5, and the secondary display 108 shows a first page 1006 running another single page application A6. Fig. 10A also shows a user input gesture 1002, namely, a spread gesture which is executed in order to launch the application manager window. Referring to Figure 10B, the device 100 is shown after the spread gesture 1002 is executed from Fig. 10A thereby revealing an application manager window or tray 1008. Within this window 1008, depictions are shown for all of the other applications and desktops that are running at the time, but are not displayed on the primary or secondary screens. These depictions of each of the applications/desktops are shown by the icons 1010, each icon representing a separate application or desktop. The application manager window or tray 1008 also illustrates each of the other running applications/desktops in their respective order within the window stacks.

[0021] One convention that is contemplated in the preferred embodiment of the present invention for displaying an order of a window stack when the window 1008 is activated is to depict each of the other applications/desktops with their respective icons 1010 placed next to one another in a linear format ftom left to right, and in which the right-most icon 1010 represents an application/desktop which is the top or lead application/desktop within the management window 1008, and the left-most icon 1010 represents an application/desktop which is the lowest or bottom application/desktop in the window stack. The icons representing the application/windows between the left and right-most icons are depicted in their respective orders as compared to the top and bottom application/desktops. Therefore, in the example of Fig. 10B, application A1 is the bottom application, A2 is the next to the bottom application, application A3 is the next application in the stack above application A2, and application A4 represents the top or lead application in the group of applications illustrated in the window 1008. Fig. 10B also illustrates the manager window 1008 as being separated by line 1014. This line represents a division of the manager window 1008 and accordingly that both the primary and secondary screens have their own separate window stack. The application/desktops displayed on the primary and secondary screens are above or on top of the right-most applications/desktops displayed in the manager's window 1008. Therefore, applications A5 and A6 are the top applications in their respective window stacks. It is also contemplated that the manager window 1008 could represent a single stack common to both the primary and secondary screens. Thus, applications A5 andA6 would still be considered as applications residing above all of the others, but that one of the applications A5 or A6 would be the leading or very top application, and the other being the next application in the stack. Either application could be designated the leading or very top application in the single window stack according to the present invention. To designate a manager window with just one window stack for both screens, the line 1014 could be removed in accordance with yet another embodiment of the invention.

[0022] The shapes, sizes, and arrangements of the icons 1010 can be modified in order to provide the most convenient user interface enabling the user to view all of the running applications and desktops. In the example of Fig. 10B, the particular arrangement and size of the icons 1010 only allow three to be shown per screen. However, it is contemplated that the icons shown in Fig. 10B could be smaller or of different shapes enabling additional icons to be displayed if other applications were running. In the event there are more than three additional applications/desktops running on any of the screens, the bottonmost non-displayed application/desktop would appear in the manager window if a user rearranged the window stack, and another application/desktop would become non-displayed.

[0023] Fig. 10B also illustrates another user gesture 1012, in the form of a drag gesture that allows the user to move the position of the selected application to the screen where the drag gesture is directed. Therefore referring to Fig. 10C, the state of the device 100 is shown after execution of the drag gesture 1012, and the application A5 is moved from the primary screen 104 to the secondary screen 108. Additionally, since the application A3 was the right-most application in the group of non-displayed applications/ desktops within the manager window 1008, this results in display of the application A3 on the primary screen 104. Accordingly the

drag gesture, executed in Fig. 10B results in the display of the application A3 and movement of the application A5 ftom the primary screen to the secondary screen. The execution of the gesture 1012 causes the manager window 1008 to close.

[0024] Fig. 10C also illustrates another spread gesture 1002 that is executed to re-launch the application manager 1008. Accordingly, Fig. 10D illustrates the revised order for the application/desktops within the application manager 1008. More specifically, Fig. 10D shows application A3 on page 1016, application A5 is shown on page 1004, application A1 is shown on the window stack for the primary screen 104 as well as application A2, and applications A6 and A4 are shown on the window stack for the secondary screen 108. It is further noted with respect to Fig. 10D that the previously displayed application A6, after being "moved off" the secondary screen 108 by the movement of the application A5, now becomes the bottommost application in the window stack for the secondary screen 108.

[0025] While specific conventions are shown with respect to how the applications/desktops are arranged in an order within a particular window stack, it shall be understood that other conventions could be applied to include other methods of illustrating the particular order of each of the applications/desktops within a window stacks. For example, instead of the right to left order within the window 1008 depicting a top to bottom arrangement of the applications or desktops, a left to right convention could be used indicating a top to bottom arrangement of the window/application. Further for example, the icons 1010 could appear in a manager window or tray that was oriented vertically on one of the screens, and the icons 1010 could appear in a column arrangement with the top or uppermost applications/desktop indicating the top applications/desktop in the stack.

[0026] Figs.10E-10H illustrate yet another embodiment of the present invention showing other ways in which a window stack may be manipulated by use of the application manager module/tool. Referring first to Fig. 10E, the device 100 in a first state shows page 1016 with application A3 and page 1006 with application A6. A spread gesture 1002 is executed. Now referring to Fig. 10F, the application manager window or tray 1008 is displayed and the user has executed yet another user gesture, such as a tap gesture 1020on a selected application/desktop. Tapping of one of the icons 1010 in this manner results in that particular application/desktop appearing on the screen ftom which the gesture was executed. Fig. 10F also illustrates the application A3 in the manager window1008 in dotted lines to indicate that that application will be moved from the primary screen 104 to a non-displayed application within the manager window 1008. Referring now to Fig. 10G, the tap gesture 1020 causes the manager window 1008 to close and the application A1 to appear on the primary screen 104 within another window 1022. Fig. 10G also illustrates the user executing another spread gesture 1002 in order to re-launch the application manager window. Therefore, referring to Fig. 10H a new order is illustrated for the window stack in which application A3 has been returned to the bottom of the window stack of the primary screen, and applications A4 and A5 still remain within the window stack of the secondary screen in the same order.

[0027] Figs. 10I -10M illustrate yet another preferred embodiment of the present invention. Referring first to Fig. 10I, the device 100 is shown after the user has launched the application manager, thereby displaying the window 1008 and the respective icons 1010. In the example of Fig. 10I, the user executes a drag gesture within the window 1008 in order to move the position of the selected application A5. More specifically, the user in Fig. 10I wishes to move the application A5 as the last application in the window stack for the secondary screen to the window stack for the primary screen 104.

[0028] Referring to Fig. 10J, it illustrates the application A5 as being moved to the stack within the primary screen and Fig. 10J also illustrates yet another drag gesture 1020 in which the user moves the application A4 into the window stack for the primary screen 104.

[0029] Now referring to Fig. 10K, this figure illustrates no applications/desktops in the manager window for the secondary screen. For the primary screen, the applications A2, A5 and A6 are shown in the manager window 1008, but not the application A1. The application A1 is not illustrated because the particular size of the icons only allows a total of three applications/desktops to be displayed in the manager window per screen. Therefore, the application A1 is still running, but is simply not displayed in the manager window 1008. As mentioned above, if it were desired to also illustrate the application A1, the size, shape, or arrangement of the icons 1010 could be adjusted in order that all running applications/desktops could be illustrated within the application manager window 1008. Finally, Fig. 10K also illustrates the user executing a tap gesture 1020, which is the command to move the application A5 in a displayed position on the primary screen 104.

[0030] Now referring to Fig. 10L, the tap gesture 1020 closes the application manager window 1008, the application A5 is displayed on page 1004 on the primary screen 104, and application A6 in window 1006 remains displayed on the secondary screen 1008.

[0031] Fig. 10L also illustrates yet another spread gesture 1002, and in response as shown in Fig. 10M, the application manager window 1008 is again displayed, along with each of the icons 1010 illustrating the updated order of the window stack. Accordingly, applications A5 and A6 are displayed on the primary and secondary screens, respectively, no applications appear in the manager window 1008 of the secondary screen, and applications A1, A3, and A4 are illustrated in the window stack of the primary screen.

[0032] Referring to Fig. 10N, this figure illustrates yet another spread gesture 1002 executed in order to launch the application manager 1008. Accordingly, referring to Fig. 10O, the application manager window 1008 is opened. Applications A1 and A2 reside in the window stack for primary screen 104, and applications A5 and A4 reside on the window stack for the secondary screen 108. Fig. 10O also illustrates a user executing yet another user input gesture, namely, a drag gesture 1040 which is used to move the application A5 ftom within the window stack on the secondary screen 108 to a displayed position on the primary screen 104. This type of drag gesture can be used to move any application/desktop in a window stack to any selected screen by simply dragging the selected application/desktop to the selected screen.

[0033] Now referring to Fig. 10P, execution of the drag gesture 1040 causes the application manager 1008 to close, and causes the application A5 on page 1004 to be displayed on the primary screen 104, while the application A6 on the page 1006 remains displayed on the secondary screen 108. Fig. 10P also illustrates a user executing yet another spread gesture 1002 in order to open the application manager 1008.

[0034] Now referring to Fig. 10Q, the application manager 1008 is again opened, and the new orders for the respective window stacks are shown. More specifically, for the primary screen 104, the applications in the stack include applications A1, A2 and A3, with application A5 being displayed on the primary screen. The application A3 that was replaced by the application A5 now appears as the next to highest application in the window stack for the primary screen; however, it should be understood that another protocol could be adopted such that the replaced application A3 could be placed at another location within the window stack. For the secondary screen 108, the window stack for this screen includes the displayed application A6, and the remaining application in this stack, namely application A4.

[0035] Now referring to Fig. 11, a simplified flow chart 1050 is provided to explain a basic sequence of events when a user launches the application manager in order to change the window stack(s) order. At the start 1052, the user determines the desired application/desktops to run, and then launches the applications, shown at block 1054. The user then must determine whether the apps are in the desired stack order, shown at block 1056. If the user believes the applications are in the desired stack order, the analysis is complete, shown at end block 1070. If the applications are not in the desired order, the user then executes a gesture to initiate the app manager, shown at block 1058. The user then views the applications as they are represented by the icons 1010 within the manager window 1008, and the user also determines a desired order, shown at block 1060. The user then executes the appropriate gesture to move a first selected application to a new position, shown at block 1062. In response to the executed gesture, the stack order changes, shown at block 1064. As discussed above with reference to Figs. 10A-10M, a user can execute a number of different gestures to move the applications into the desired order, such as a drag gesture on the screens 104 and 108, or tap or drag gestures within the application manager window 1008. At block 1068, The user then re-evaluates the applications in the stack and determines whether they are in the desired stack order after execution of the first move gesture, shown at block 1068. If the applications are now in the desired stack order, then the analysis is complete. If the applications are not yet in the desired stack order, the user then executes a second gesture to move another application to another location within the stack. The gestures are repeated until the user is satisfied with the desired window stack(s) order(s).

[0036] Based on the description of Figs. 10A-10Q, it is evident that the display of windows in a window stack can be selectively changed by simple gesture interactions, thereby making desktops and multiple applications available to the user with minimum effort. Each screen has a logical representation of the windows that have been opened and displayed on the screen. The logical representation is the stack that is made up of one or more windows that is arranged in an order that can be selected and changed by the user. By opening of the window manager tool, the windows can be indexed or shifted to place the window stack in a desired order as well as to move windows between separate window stacks, such as separate window stacks that may be present on both the primary and secondary screens. The ordered arrangement of the stack

maintains the logical construction of the stack intact when the user navigates the window stack(s). The user can open the manager window 1008 repeatedly in order to continue to change the order of the window stack(s), and each subsequent gesture executed is based on the present order of the window stack(s) prior to executing the subsequent gesture(s). Therefore, a logical order is maintained for each window stack at all times. The use of the icons 1010 within the window manager provides a very convenient visual representation of the running applications/desktops. Therefore the device operates in a very intuitive way for the user to easily manage multiple desktops and applications within a dual screen environment.

Claims

1. A method of displaying a plurality of desktops and/or applications, each having at least one window, on a multi-screen device, comprising:

receiving, by a processor, a first input that represents an instruction to reveal one of a desktop or application on a first display of the multi-screen device;

selecting, by a processor, a first desktop or application to display on the first display;

receiving, by a processor, a second input that represents an instruction to reveal one of a desktop or application on a second display of the multi-screen device;

selecting, by a processor, a second desktop or application to display on the second display;

displaying , by a processor, the selected first and second desktops or applications on the first and second displays;

receiving, by a processor, a third input that represents an instruction to launch an application manager feature ;

displaying, by a processor, the applications or desktops on the first and second displays, and displaying a management window showing all other applications or desktops currently running;

determining, by a processor, a desired order for a window stack comprising the first and second applications or desktops, and said other applications or desktops;

receiving, by a processor, a fourth input that represents an instruction to selectively move one of said other applications or desktops within the window stack to a displayed position on said first or second displays;

displaying, by a processor, the window stack in a new order in response to said fourth input.

2. The method of claim 1, wherein the fourth input is a user gesture comprising at least one of a flick or drag gesture.

3. The method of claim 1, wherein the third input is a user gesture comprising a spread gesture, and wherein the spread gesture is executed on off-screen areas of said first and seconds displays with one finger of a user placed on one display, and another finger of the user placed on the other display.

4. The method of claim 1, wherein said receiving a fourth input step is executed a number of times until the window stack is displayed in the desired order.

5. The method of claim 1, wherein at least one management window appears on said first and second displays, and the other applications or desktops are arranged linearly within the management window from lefr to right, and the lefr to right order of the other applications or desktops corresponds to an order of the other applications or desktops within the window stack.

6. The method of claim 1, wherein the fourth input is a drag gesture executed on a selected one of said other applications or desktops, and the drag is pointed toward one of said first or second displays for displaying the selected one on the display to which the drag gesture is pointed.

7. The method of claim 5, wherein separate management windows appear on said first and second displays denoting a first window stack for the first display and a second window stack for the second display, and wherein the other applications or desktops are arranged linearly within each management window from left to right, and the lefr to right order of the other applications or desktops correspond to an order of the other applications or desktops within the window stack for each corresponding display.

8. A non-transitory computer-readable medium having stored thereon instructions that cause a computing system to execute a method, the instructions comprising:

instructions configured to receive a first predetermined input that represents an instruction to determine and reveal a first desktop or application on a first display of the multi-screen device;

instructions configured to receive a second predetermined input that represents an instruction to determine and reveal a second desktop or application on a second display of the multi-screen device;

instructions configured to respond to the first and second predetermined inputs with outputs that cause the first desktop or application to be displayed on the first display and cause the second desktop or application to be displayed on the second display

instructions configured to receive a third predetermined input that represents an instruction to launch an application manager feature;

instructions configured to respond to the third predetermined input that causes a display of a management window showing all other applications or desktops currently running;

instructions configured to receive a fourth predetermined input that represents an instruction to selectively move one of said other applications or desktops within the window stack to a displayed position on said first or second displays; and

instructions configured to respond to the fourth predetermined input that causes displaying of the window stack in a new order.

9. The computer-readable medium of claim 8, wherein said fourth predetermined input step is executed a number of times by a user until the window stack is displayed in the desired order.

10. The computer-readable medium of claim 8, wherein said management window appears on said first and second displays, and the other applications or desktops are arranged linearly within the management window from left to right, and the left to right order of the other applications or desktops corresponds to an order of the other applications or desktops within the window stack.

11. The computer-readable medium of claim 8 wherein the fourth input is a drag gesture executed on a selected one of said other applications or desktops, and the drag is pointed toward one of said first or second displays for displaying the selected one on the display to which the drag gesture is pointed.

12. A multi-screen user device, comprising:

a first display including a first display area;

a second display including a second display area;

a first user input gesture area of the first display;

a second user input gesture area of the second display, wherein the first and second user input gesture areas are configured to accept input from a user.

a computer-readable medium having instructions stored thereon that include:

a first set of instructions configured to determine a number and identity of desktops or applications selected to be run by a user;

a second set of instructions configured to determine, based on the number and identity of the desktops or applications running, a window stack comprising a logical representation of the desktops and applications within an ordered group whereby a user can selectively move a selected desktop or application to either a visible position on another display or a non-visible position; and

a third set of instructions responsive to a gesture made by the user that represents an instruction to launch an application manager feature, wherein the applications or desktops are displayed on the first and second displays, and a management window is displayed showing all other applications or desktops currently running;

a fourth set of instructions responsive to one or more gestures made by the user that represent an instruction to selectively move one of said other applications or desktops within the window stack to a displayed position on said first or second displays; and

wherein the window stack is displayed in a new order in response to said fourth set of instructions.

13. The device of claim 12, wherein said fourth set of instructions are executed a number of times until the window stack is displayed in the desired order.

14. The device of claim 12, wherein at least one management window appears on said first and second displays, and the other applications or desktops are arranged linearly within the management window from lefr to right, and the lefr to right order of the other applications or desktops corresponds to an order of the other applications or desktops within the window stack

15. The device of claim 12, wherein separate management windows appear on said first and second displays denoting a first window stack for the first display and a second window stack for the second display, and wherein the other applications or desktops are arranged linearly within each management window from left to right, and the lefr to right order of the other applications or desktops correspond to an order of the other applications or desktops within the window stack for each corresponding display.

16. The device of claim 12, wherein the fourth input is a drag gesture executed on a selected one of said other applications or desktops, and the drag is pointed toward one of said first or second displays for displaying the selected one on the display to which the drag gesture is pointed.

ABSTRACT

A dual-screen user device and methods for revealing a combination of desktops on single and multiple screens are disclosed. Selected desktops and/or running applications are displayed on dual screen displays. Desktops and applications can be shifted between screens by user gestures, and/or moved off of the screens and therefore hidden. Hidden desktops and screens can be re-displayed by yet other gestures. The desktops and applications are arranged in a window stack that represents a logical order of the desktops and applications providing a user with an intuitive ability to manage multiple applications/desktops running simultaneously. One user gesture launches an applications management window that provides visual indications of all of the applications and desktops running at the time. Other gestures can rearrange the order of all of the applications and desktops in the window stack. One particular gesture allows a user to drag a selected application or desktop appearing in the applications management window to a selected screen.

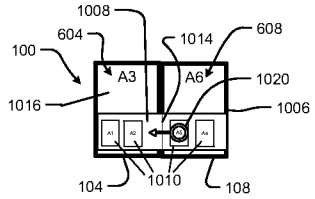


FIG. 10I

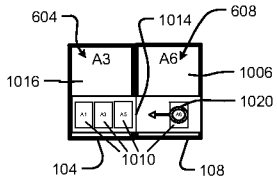


FIG. 10J

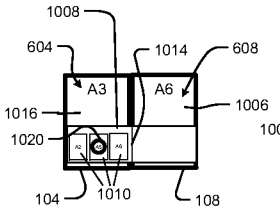


FIG. 10K

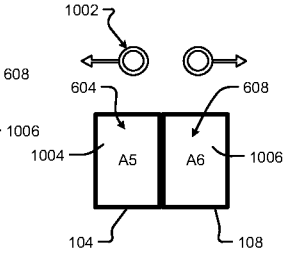


FIG. 10L

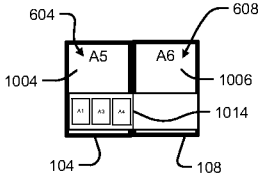


FIG. 10M

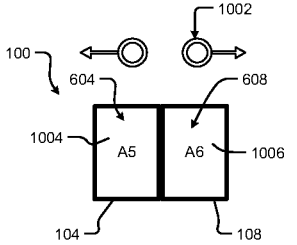


FIG. 10A

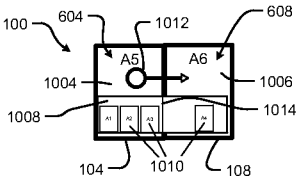


FIG. 10B

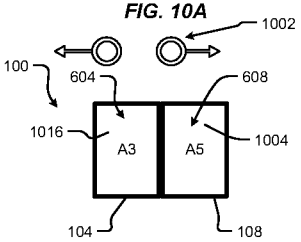


FIG. 10C

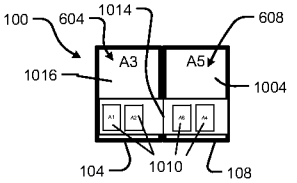


FIG. 10D

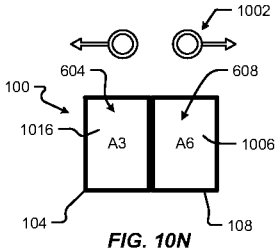


FIG. 10N

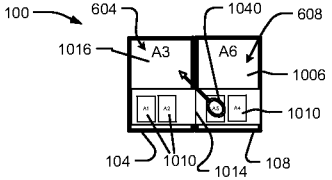


FIG. 10O

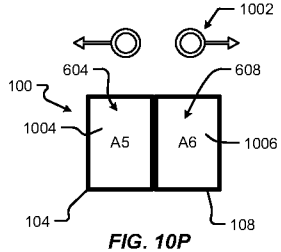


FIG. 10P

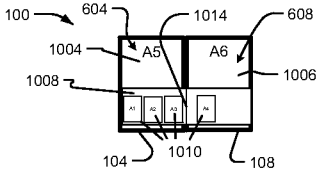


FIG. 10Q

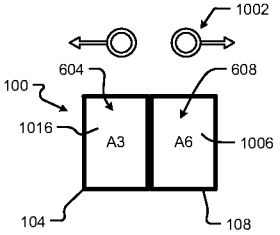


FIG. 10E

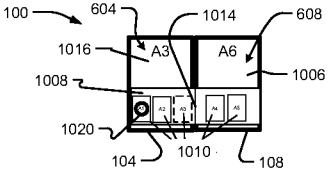


FIG. 10F

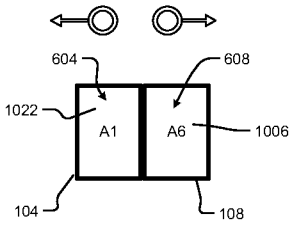


FIG. 10G

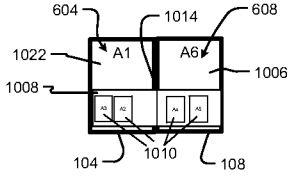


FIG. 10H

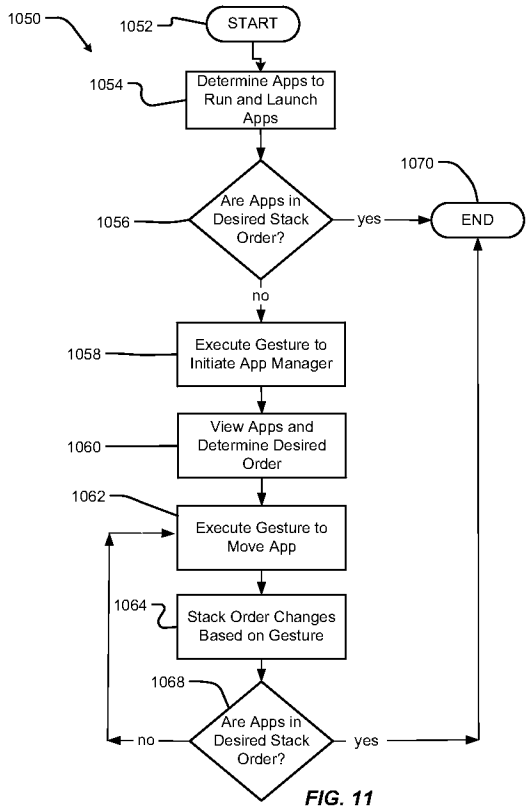

1050
1052
START
1054
Determine Apps to Run and Launch Apps
1056
Are Apps in Desired Stack Order?
yes
1070
END
no
1058
Execute Gesture to Initiate App Manager
1060
View Apps and Determine Desired Order
1062
Execute Gesture to Move App
1064
Stack Order Changes Based on Gesture
1068
Are Apps in Desired Stack Order?
no
yes

FIG. 11

OPENING CHILD WINDOWS IN DUAL DISPLAY COMMUNICATION DEVICES

[0100] The device 100 manages desktops and/or windows with at least one window stack 1700, 1728, as shown in Figs. 9A and 9B. A window stack 1700, 1728 is a logical arrangement of active and/or inactive windows for a multi-screen device. For example, the window stack 1700, 1728 may be logically similar to a deck of cards, where one or more windows or desktops are arranged in order, as shown in Figs. 9A and 9B. An active window is a window that is currently being displayed on at least one of the touch sensitive displays 110, 114. For example, windows 104 and 108 are active windows and are displayed on touch sensitive displays 110 and 114. An inactive window is a window that was opened and displayed but is now "behind" an active window and not being displayed. In embodiments, an inactive window may be for an application that is suspended, and thus, the window is not displaying active content. For example, windows 1712, 1716, 1720, and 1724 are inactive windows.

[0101] A window stack 1700, 1728 may have various arrangements or organizational structures. In the embodiment shown in Fig. 9A, the device 100 includes a first stack 1760 associated with a first touch sensitive display 110 and a second stack associated with a second touch sensitive display 114. Thus, each touch sensitive display 110, 114 can have an associated window stack 1760, 1764. These two window stacks 1760, 1764 may have different numbers of windows arranged in the respective stacks 1760, 1764. Further, the two window stacks 1760, 1764 can also be identified differently and managed separately. Thus, the first window stack 1760 can be arranged in order from a first window 1704 to a next window 1720 to a last window 1724 and finally to a desktop 1722, which, in embodiments, is at the "bottom" of the window stack 1760. In embodiments, the desktop 1722 is not always at the "bottom" as application windows can be arranged in the window stack below the desktop 1722, and the desktop 1722 can be brought to the "top" of a stack over other windows during a desktop reveal. Likewise, the second stack 1764 can be arranged from a first window 1708 to a next window 1712 to a last window 1716, and finally to a desktop 1718, which, in embodiments, is a single desktop area, with desktop 1722, under all the windows in both window stack 1760 and window stack 1764. A logical data structure for managing the two window stacks 1760, 1764 may be as described in conjunction with Fig. 10.

[0102] Another arrangement for a window stack 1728 is shown in Fig. 9B. In this embodiment, there is a single window stack 1728 for both touch sensitive displays 110, 114.

a function of the orientation of the device 100, the context of what programs, functions, software, etc. are being executed on the device 100, how the stack is positioned when the new window is opened, etc. To insert the window in the stack, the position in the stack for the window is determined and the touch sensitive display 110, 114 to which the window is associated may also be determined. With this information, a logical data structure for the window can be created and stored. When user interface or other events or tasks change the arrangement of windows, the window stack(s) can be changed to reflect the change in arrangement. It should be noted that these same concepts described above can be used to manage the one or more desktops for the device 100.

[0106] A logical data structure 1800 for managing the arrangement of windows or desktops in a window stack is shown in Fig. 10. The logical data structure 1800 can be any data structure used to store data whether an object, record, file, etc. The logical data structure 1800 can be stored in any type of database or data storage system, regardless of protocol or standard. In embodiments, the logical data structure 1800 includes one or more portions, fields, attributes, etc. that store data in a logical arrangement that allows for easy storage and retrieval of the information. Hereinafter, these one or more portions, fields, attributes, etc. shall be described simply as fields. The fields can store data for a window identifier 1804, dimensions 1808, a stack position identifier 1812, a display identifier 1816, and/or an active indicator 1820. Each window in a window stack can have an associated logical data structure 1800. While only a single logical data structure 1800 is shown in Fig. 10, there may be more or fewer logical data structures 1800 used with a window stack (based on the number of windows or desktops in the stack), as represented by ellipses 1824. Further, there may be more or fewer fields than those shown in Fig. 10, as represented by ellipses 1828.

[0107] A window identifier 1804 can include any identifier (ID) that uniquely identifies the associated window in relation to other windows in the window stack. The window identifier 1804 can be a globally unique identifier (GUID), a numeric ID, an alphanumeric ID, or other type of identifier. In embodiments, the window identifier 1804 can be one, two, or any number of digits based on the number of windows that can be opened. In alternative embodiments, the size of the window identifier 1804 may change based on the number of windows opened. While the window is open, the window identifier 1804 may be static and remain unchanged.

Thus, the window stack 1728 is arranged from a desktop 1758 to a first window 1744 to a last window 1756. A window can be arranged in a position among all windows without an association to a specific touch sensitive display 110, 114. In this embodiment, a window is in the order of windows. Further, at least one window is identified as being active. For example, a single window may be rendered in two portions 1732 and 1736 that are displayed on the first touch sensitive screen 110 and the second touch sensitive screen 114. The single window may only occupy a single position in the window stack 1728 although it is displayed on both displays 110, 114.

[0103] Yet another arrangement of a window stack 1760 is shown in Figs. 9C through 9E. The window stack 1760 is shown in three "elevation" views. In Fig. 9C, the top of the window stack 1760 is shown. Two sides of the window stack 1760 are shown in Figs. 9D and 9E. In this embodiment, the window stack 1760 resembles a stack of bricks. The windows are stacked on each other. Looking from the top of the window stack 1760 in Fig, 9C, only the top most windows in the window stack 1760 are seen in different portions of the composite display 1764. The composite display 1764 represents a logical model for the entire display area of the device 100, which can include touch sensitive display 110 and touch sensitive display 114. A desktop 1786 or a window can occupy part or all of the composite display 1764.

[0104] In the embodiment shown, the desktop 1786 is the lowest display or "brick" in the window stack 1760. Thereupon, window 1 1782, window 2 1782, window 3 1768, and window 4 1770 are layered. Window 1 1782, window 3 1768, window 2 1782, and window 4 1770 only occupy a portion of the composite display 1764. Thus, another part of the stack 1760 includes window 8 1774 and windows 5 through 7 shown in section 1790. Only the top window in any portion of the composite display 1764 is actually rendered and displayed. Thus, as shown in the top view in Fig. 9C, window 4 1770, window 8 1774, and window 3 1768 are displayed as being at the top of the display in different portions of the window stack 1760. A window can be dimensioned to occupy only a portion of the composite display 1760 to "reveal" windows lower in the window stack 1760. For example, window 3 1768 is lower in the stack than both window 4 1770 and window 8 1774 but is still displayed. A logical data structure to manage the window stack can be as described in conjunction with Fig. 10.

[0105] When a new window is opened, the newly activated window is generally positioned at the top of the stack. However, where and how the window is positioned within the stack can be

[0108] Dimensions 1808 can include dimensions for a window in the composite display 1760. For example, the dimensions 1808 can include coordinates for two or more corners of the window or may include one coordinate and dimensions for the width and height of the window. These dimensions 1808 can delineate what portion of the composite display 1760 the window may occupy, which may the entire composite display 1760 or only part of composite display 1760. For example, window 4 1770 may have dimensions 1880 that indicate that the window 1770 will occupy only part of the display area for composite display 1760, as shown in Figs. 9C through 9E. As windows are moved or inserted in the window stack, the dimensions 1808 may change.

[0109] A stack position identifier 1812 can be any identifier that can identify the position in the stack for the window or may be inferred from the window's control record within a data structure, such as a list or a stack. The stack position identifier 1812 can be a GUID, a numeric ID, an alphanumeric ID, or other type of identifier. Each window or desktop can include a stack position identifier 1812. For example, as shown in Fig. 9A, window 1 1704 in stack 1 1760 can have a stack position identifier 1812 of 1 identifying that window 1704 is the first window in the stack 1760 and the active window. Similarly, window 6 1724 can have a stack position identifier 1812 of 3 representing that window 1724 is the third window in the stack 1760. Window 2 1708 can also have a stack position identifier 1812 of 1 representing that window 1708 is the first window in the second stack 1764. As shown in Fig. 9B, window 1 1744 can have a stack position identifier 1812 of 1, window 3, rendered in portions 1732 and 1736, can have a stack position identifier 1812 of 3, and window 6 1756 can have a stack position identifier 1812 of 6. Thus, depending on the type of stack, the stack position identifier 1812 can represent a window's location in the stack.

[0110] A display identifier 1816 can identify that the window or desktop is associated with a particular display, such as the first display 110 or the second display 114, or the composite display 1760 composed of both displays. While this display identifier 1816 may not be needed for a multi-stack system, as shown in Fig. 9A, the display identifier 1816 can indicate whether a window in the serial stack of Fig. 9B is displayed on a particular display. Thus, window 3 may have two portions 1732 and 1736 in Fig. 9B. The first portion 1732 may have a display identifier 1816 for the first display while the second portion 1736 may have a display identifier 1816 for the second display 114. However, in alternative embodiments, the window may have

two display identifier 1816 that represent that the window is displayed on both of the displays 110, 114, or a display identifier 1816 identifying the composite display. In another alternate embodiment, the window may have a single display identifier 1816 to represent that the window is displayed on both of the displays 110, 114.

[0111] Similar to the display identifier 1816, an active indicator 1820 may not be needed with the dual stack system of Fig. 9A, as the window in stack position 1 is active and displayed. In the system of Fig. 9B, the active indicator 1820 can indicate which window(s) in the stack is being displayed. Thus, window 3 may have two portions 1732 and 1736 in Fig. 9. The first portion 1732 may have an active indicator 1820 while the second portion 1736 may also have an active indicator 1820. However, in alternative embodiments, window 3 may have a single active indicator 1820. The active indicator 1820 can be a simple flag or bit that represents that the window is active or displayed.

[0112] An embodiment of a method 900 for creating a window stack is shown in Fig. 11. While a general order for the steps of the method 1900 is shown in Fig. 11. Generally, the method 1900 starts with a start operation 1904 and ends with an end operation 1928. The method 1900 can include more or fewer steps or can arrange the order of the steps differently than those shown in Fig. 11. The method 1900 can be executed as a set of computer-executable instructions executed by a computer system and encoded or stored on a computer readable medium. Hereinafrer, the method 1900 shall be explained with reference to the systems, components, modules, software, data structures, user interfaces, etc. described in conjunction with Figs. 1-10.

[0113] A multi-screen device 100 can receive activation of a window, in step 1908. In embodiments, the multi-screen device 100 can receive activation of a window by receiving an input from the touch sensitive display 110 or 114, the configurable area 112 or 116, a gesture capture region 120 or 124, or some other hardware sensor operable to receive user interface inputs. The processor may execute the Task Management Module 540 may receive the input. The Task Management Module 540 can interpret the input as requesting an application task to be executed that will open a window in the window stack.

[0114] In embodiments, the Task Management Module 540 places the user interface interaction in the task stack 552 to be acted upon by the Display Configuration Module 568 of the Multi-Display Management Module 524. Further, the Task Management Module 540 waits

for information from the Multi-Display Management Module 524 to send instructions to the Window Management Module 532 to create the window in the window stack.

[0115] The Multi-Display Management Module 524, upon receiving instruction from the Task Management Module 540, determines to which touch portion of the composite display 1760, the newly activated window should be associated, in step 1912. For example, window 4 1770 is associated with the a portion of the composite display 1764 In embodiments, the device state module 574 of the Multi-Display Management Module 524 may determine how the device is oriented or in what state the device is in, e.g., open, closed, portrait, etc. Further, the preferences module 572 and/or requirements module 580 may determine how the window is to be displayed. The gesture module 576 may determine the user's intentions about how the window is to be opened based on the type of gesture and the location of where the gesture is made.

[0116] The Display Configuration Module 568 may use the input from these modules and evaluate the current window stack 1760 to determine the best place and the best dimensions, based on a visibility algorithm, to open the window. Thus, the Display Configuration Module 568 determines the best place to put the window at the top of the window stack 1760, in step 1916. The visibility algorithm, in embodiments, determines for all portions of the composite display, which windows are at the top of the stack. For example, the visibility algorithm determines that window 3 1768, window 4 1770, and window 8 1774 are at the top of the stack 1760 as viewed in Figs. 9C through 9E. Upon determining where to open the window, the Display Configuration Module 568 can assign a display identifier 816 and possibly dimensions 808 to the window. The display identifier 816 and dimensions 808 can then be sent back to the Task Management Module 540. The Task Management Module 540 may then assign the window a stack position identifier 812 indicating the windows position at the top of the window stack.

[0117] In embodiments, the Task Management Module 540 sends the window stack information and instructions to render the window to the Window Management Module 532. The Window Management Module 532 and the Task Management Module 540 can create the logical data structure 800, in step 1924. Both the Task Management Module 540 and the Window Management Module 532 may create and manage copies of the window stack. These copies of the window stack can be synchronized or kept similar through communications between the Window Management Module 532 and the Task Management Module 540. Thus,

the Window Management Module 532 and the Task Management Module 540, based on the information determined by the Multi-Display Management Module 524, can assign dimensions 808, a stack position identifier 812 (e.g., window 1 1782, window 4 1770, etc.), a display identifier 816 (e.g., touch sensitive display 1 110, touch sensitive display 2 114, composite display identifier, etc,), and an active indicator 820, which is generally always set when the window is at the "top" of the stack. The logical data structure 800 may then be stored by both the Window Management Module 532 and the Task Management Module 540. Further, the Window Management Module 532 and the Task Management Module 540may thereinafrer manage the window stack and the logical data structure(s) 800.

[0118] FIG. 12 depicts a further window stacking configuration. A plurality of windows 1, 2, 3, 4, 5, 6, 7, and 8, whether from the same or different multi-screen or single-screen applications, are depicted. The touch sensitive display 110 currently has window 4 in the active display position while touch sensitive display 114 currently has window 5 in the active display position. The stack for touch sensitive display 110, from top to bottom, has window 4 in the active display position and windows 3, 2, and 1 positioned behind it. The stack for touch sensitive display 114, from top to bottom, has window 5 in the active display position and windows 6, 7, and 8 positioned behind it.

[0119] The desktops D1, D2, D3, D4, D5, and D6 are positioned behind the window stacks. The desktops can be viewed as a desktop stack different from the window stack. Viewed in this manner, the touch sensitive display 110 has a corresponding desktop stack comprising desktops D3, D2, and D1,with desktop D1 being at the bottom 2300 stack position and with desktop D3 at the top stack position being capable of being displayed with window 4 (depending on the window position and size (whether maximized or minimized), and the touch sensitive display 114 has a corresponding desktop stack has a corresponding desktop stack comprising desktops D4, D5, and D6,with desktop D6 being at the bottom 2304 stack position and with desktop D4 at the top stack position being capable of being displayed with window 5 (depending on the window position and size (whether maximized or minimized). Conceptually, the desktops can be viewed as a canvas divided, in this example, into six sections of which two can be displayed, at any one time, the touch sensitive displays 110, 114. When the device 100 is in the closed state, this conceptual model, in one configuration, persists. In this configuration, only one window and desktop stacks can be seen (that corresponding to the primary screen) but the other

window and desktop stacks are virtual; that is, they are maintained in memory but cannot be seen because the secondary screen is not enabled.

[0120] Single-screen applications are typically related only to one window while multi-screen applications can be related to one or more windows. The latter hierarchical relationship is depicted in Fig. 8. As shown in Fig. 8, first window 1 is a root window, second and third windows 1.1 and 1.1.1 are node windows, and fourth and fifth windows 1.1.1.1 and 1.1.2 are leaf windows. The windows are in a hierarchical tree-like sequence and are controlled by a common multi-screen application. The root first window 1 may be a top level view of the hierarchical application window sequence such that there is no parent window corresponding to the root window. The root first window 1 may be a parent to a node window, such as second window 1.1 and the second window 1.1 is a child of the first window 1. One or more node windows may be provided that are related as parent/children. A node window may also serve as a leaf window. By leaf window, it is meant that the leaf window has no corresponding node screen for which the leaf window is a parent (or a leaf window has no child windows). As such, the leaf window does not have any children node windows, Fourth and fifth windows 1.1.1.1 and 1.1.2 are examples of leaf windows. The root view refers to all of the windows controlled by a multi-screen application at a point in time.

[0121] A number of examples will now be discussed with reference to minimizing and maximizing root view windows and subcomponents thereof. Prior to discussing the examples, some fundamental rules regarding maximizing and minimizing displayed images will be discussed. A maximize or minimize operation is typically initiated by a user gesture, particularly a flip or drag. In a drag gesture, a 1:1 map reaction between the user's digit and the displayed image is used as the user "drags" the displayed image to the other touch sensitive display. The user typically releases the displayed image before it is fully moved to the other touch sensitive display. The initial speed after release is generally decreased progressively or gradually until the displayed image is fully located in the desired position. Too little displacement of the displayed image generally will not cause the image to be relocated. Such a low level of displacement is assumed to be accidental or incidental to the user using the device.

[0122] In a root view, or set of hierarchically related windows, the user, when the device is in dual screen mode, can maximize the related windows of the multi-screen application to enable side-by-side viewing of the related windows (e.g., the parent and child windows), thereby

viewing the window hierarchy as a linear progression, with the top-most window being displayed on the left and the lowest-most or deepest level window being displayed on the right (or vice versa). As discussed below, this visual configuration can be controlled using directional transitions conveying a navigation direction and location of a window in relation to a launching point of the window.

[0123] A multi-screen application window generally opens in minimized mode. A continuing drag and release gesture when a window of a multi-screen application, originally minimized in a first touch sensitive display, is at least half-way across the second touch sensitive display will generally cause the window to be maximized, or displayed in both the first and second touch sensitive displays. Dragging a window of a multi-screen application originally minimized in a first touch sensitive display to a second touch sensitive display all the way across to the far edge of the second touch sensitive display will generally maximize the window across both the first and second touch sensitive displays. When the user continues to drag in the same direction after the window of the multi-screen application is expanded fully (or maximized across both the first and second touch sensitive displays and touching both far (non-adjacent) edges of the first and second touch sensitive displays), the window will pull away from the far edge of the first touch sensitive display and, once the far edge of the window is half-way across the first touch sensitive display, the application window will generally reposition in minimized mode in the second touch sensitive display.

[0124] The flick gesture will not generally maximize a window of a multi-screen application but generally will reposition a minimized window of a single- or multi-screen application. The flick gesture will generally not minimize window(s) of a maximized application. The flick gesture is a way to move rapidly through the windows of a stack, whether the windows are of a single- or multi-screen application.

[0125] A first example will be discussed with reference to Fig. 13, a user selects an application shortcut on a first desktop D1 to launch a multi-screen application, such as Outlook™. The application opens to display a defined parent or node second window 1.1 (1304) in the screen or display of the first desktop D1 (or the screen from which the application was launched). The user, by a drag gesture 1302, expands or maximizes the application to both the first and second touch sensitive displays to display the first window 1.1 (1304) and the node third window 1.1.1 (1308). Once expanded, the node third window 1.1.1 (1308) appears based on rules set by the

particular application, *e.g.,* the node third window 1.1.1 (1308) selected is at a selected position and/or relationship to the parent second window 1.1 (1304). The user selects, such as by a tap gesture 1316, the node third window 1.1.1 (1308) towards the other first touch sensitive display110 (displaying the parent second window 1.1 (1304), causing the parent second window 1.1 (1304) to slide off-display and the leaf fifth window 1.1.1.1 (1312) to appear in the second touch sensitive display. If, on the other hand, the user selects, such as by a tap gesture 1316, the parent second window 1.1 (1304) moves to and is displayed by the second touch sensitive display 114, the node third window 1.1.1 (1308) slides to the right off the second touch sensitive display 114 to an inactive display position, and a first window 1 (1300) at a hierarchy level higher than the parent second window 1.1(1304) moves into and is displayed by the first touch sensitive display 110.

[0126] A second example will be applied to Figs. 14A-C. The first and second touch sensitive displays 110, 114 display, respectively, a leaf view fifth window 1.1.1.1 (1312) and a second desktop D2 (Fig. 14A). Stated another way, the first touch sensitive display 110 has the leaf view fifth window 1.1.1.1 (1312) in an active display position, and the second touch sensitive display 114 has the second desktop D2 in an active display position. In a gesture capture region 120 or 124 (or in other configurations the first or second touch sensitive display 110 or 114) receives, from the user, a gesture 1302. Exemplary gestures include a drag or flick to the right(though one or more other gestures may be employed). By the gesture 1302, the user seeks to maximize the root view to show the node third window 1.1.1 (1308) and the leaf fifth window 1.1.1.1 (1312) in separate displays (Fig. 14B). As a general rule, a leaf window (or window having no child) is positioned on the second touch sensitive display 114. Thus, the leaf fifth window 1.1.1.1 (1312) moves to the active display position on the second touch sensitive display 114, and the node third window 1.1.1 (1308) is moved to the active display position on the first touch sensitive display 110 as shown by Fig. 14C. The second desktop D2, meanwhile, moves to an inactive display position on the second touch sensitive display 114. The leaf fifth window 1.1.1.1 (1312) does this by sliding over to the second touch sensitive display 114, thereby revealing or uncovering the node third window 1.1.1 (1308). As will be appreciated, a reverse general rule could also be applied; that is, the leaf window could be positioned on the first touch sensitive display 110.

[0127] A third example will be discussed with reference to Figs. 15A-C in which the first touch sensitive display 110 has a first desktop D1 in an active display position, and the second touch sensitive display 114 has the leaf view fifth window 1.1.1.1 (1312) in an active display position. In a gesture capture region 120 or 124 (or in other configurations the first or second touch sensitive display 110 or 114) receives, from the user, a gesture 1302, such as a flick or drag, to the left. By the gesture 1302, the user seeks to maximize the root view to show the node third window 1.1.1 (1308) and the leaf fifth window 1.1.1.1 (1312) in separate displays (Fig. 15B). As a general rule, a leaf window (or window having no child) is positioned on the second touch sensitive display 114. Thus, the leaf fifth window 1.1.1.1 (1312) remains in the active display position on the second touch sensitive display 114, and the node third window 1.1.1 (1308) is moved to the active display position on the first touch sensitive display 110 as shown by Fig. 15C. The first desktop D1, meanwhile, moves to an inactive display position on the first touch sensitive display 110. The node third window 1.1.1 (1308) does this by sliding from under the leaf fifth window 1.1.1.1 (1312). As will be appreciated, a reverse general rule could also be applied; that is, the leaf window could be positioned on the first touch sensitive display 110.

[0128] A fourth example will be discussed with reference to Figs. 16A-C in which the first touch sensitive display 110 has the node second window 1.1 (1304) in an active display position (and the node and leaf third and fifth windows 1.1.1 (1308) and 1.1.1.1 (1312) in inactive display positions), and the second touch sensitive display 114 has the second desktop D2 in an active display position. In a gesture capture region 120 or 124 (or in other configurations the first or second touch sensitive display 110 or 114) receives, from the user, a gesture 1302, such as a flick or drag, to the right. By the gesture 1300, the user seeks to maximize the root view to show the node second window 1.1 (1304) and node third window 1.1.1 (1308) in separate displays (Fig. 16B). As a general rule, whenever a child window (e.g., a node or leaf window) is uncovered through maximization, it is uncovered on the second touch sensitive display 114. Thus, the node second window 1.1 (1304) remains in the active display position on the first touch sensitive display 110, and the node third window 1.1.1 (1308) is moved to the active display position on the second touch sensitive display 114 as shown by Fig. 16C. The second desktop D2, meanwhile, moves to an inactive display position on the second touch sensitive display 114. The node third window 1.1.1 (1308) does this by sliding out from under the second node window 1.1

(1304). As will be appreciated, a reverse general rule could also be applied; that is, the child window could be positioned on the first touch sensitive display 110.

[0129] A fifth example will be discussed with reference to Figs. 17A-C in which the first touch sensitive display 110 has the first desktop D1 in an active display position, and the second touch sensitive display 114 has the node second window 1.1 (1304) in an active display position (and the node and leaf third and fifth windows 1.1.1 (1308) and 1.1.1.1 (1312) in inactive display positions). In a gesture capture region 120 or 124 (or in other configurations the first or second touch sensitive display 110 or 114) receives, from the user, a gesture 1302, such as a flick or drag, to the left. By the gesture 1302, the user seeks to maximize the root view to show the node second window 1.1 (1304) and node third window 1.1.1 (1308) in separate displays (Fig. 17B). Thus, the node third window 1.1.1 (1308) moves to the active display position on the second touch sensitive display 114, and the node second window 1.1 (1304) is moved to the active display position on the first touch sensitive display 110 as shown by Fig. 17C. The first desktop D1, meanwhile, moves to an inactive display position on the first touch sensitive display 110. The node second window 1.1 (1304) does this by sliding over from above the node third window 1.1.1 (1308).

[0130] A sixth example will be discussed with reference to Fig. 18. A user has opened an email application, such as by selecting a shortcut in the first desktop D1, in the first touch sensitive display110 while the second touch sensitive display 114 displays the second desktop D2. The email application opens at the default "starting point", which is the email inbox window 1800 shown in Figs. 18A-B. As shown in Fig. 18B, the user inputs a drag gesture to the right in the off-display gesture capture region 120 to maximize the email application window. The inbox window remains displayed in the first touch sensitive display 110 and a child window 1804 (which is similar to third node window 1.1.1 (1308)) opens in the second touch sensitive display 114. In one configuration, the first email is displayed by default in the child window 1804. The user can select another email to open in a new child window in the second touch sensitive display 114 by a tap gesture received by the email inbox window 1800 in the first touch sensitive display 110. The tap gesture may be received at a specific email's location to cause that email to be displayed in a child window on the second touch sensitive display 114. A tap-selectable affordance may be provided for each displayed email to select that specific email for display in a child window. An account folder window (not shown), which is similar to first

parent window 1 (1300) can be selected by the user, thereby moving the email inbox window (which is similar to second node window 1.1 (1304) to the second touch sensitive display 114. If the user were then, by a tap gesture on the inbox window now displayed on the second touch sensitive display, to select an email for viewing, the inbox window moves to the first touch sensitive display 110 for display, causing the account folder window to move off to the left of the first touch sensitive display 110, and the selected email is displayed in a child window on the second touch sensitive display 114.

[0131] The above example can apply to other itemized listings, such as contacts, instant messages, voice mail, and the like.

[0132] Conventional single or dual-screen devices allow a user to access individual emails or other "child" information from an application by opening new windows within an application. Typically, these new child windows open over the parent windows from which they originated. When a user wishes to return to a parent window (like an email list, for example) it is necessary to work back through all of the opened child windows. The above examples show that the overall hierarchy and context of parent-child relationships with a multi-screen application can be preserved by launching each lower level or child displayed image in an additional secondary window rather than simply loading child content in the same window.

[0133] As shown by the above examples, the device 1900 can allow a user to open and, in some configurations, close or move automatically child windows with a simple linear drag motion across the gesture capture region 120 or 124. This motion can eliminate the need to work through opened child windows to access the parent window and allow the display area to remain unblocked during the creation of a child window. This off-display gesture capture region 120 or 124 can allow a user to quickly and easily open child windows of applications, thereby saving time and substantially minimizing required user input for accessing data. The end result can be a very intuitive operation of the phone for many users.

[0134] In the various examples, middleware 520, particularly one or more of the Multi-Display Management (MDM) class 524, a Surface Cache class 528, a Window Management class 532, an Activity Management class 536, and an Application Management class 540, independently or collectively, receive a command to manipulate a displayed set of related windows from a multi-screen application in a predetermined manner, such as by detecting a gesture (step 1900). In response, middleware 520 determines whether the set of related windows is currently maximized

(in which case each of the touch sensitive displays 110 and 114 display one of the related windows) or minimized (in which case only one of the touch sensitive displays one of the windows) (step 1904). In decision diamond 1908, the middleware 520 decides how to process the command based on whether the window(s) is maximized or minimized. When maximized, the middleware 520 determines whether the device 100 is in the portrait or landscape display mode and, when in the landscape display mode, which of the touch sensitive displays 110 and 114 is on top and which is on the bottom (step 1912). When in portrait display mode, the middleware 520 causes the lower level window (*e.g.*, leaf window 1.1.1.1) to be displayed on a predetermined one of the left or right touch sensitive display. When in landscape display mode, the middleware 520 causes the lower level window to be displayed on a predetermined one of the upper or lower touch sensitive display. When the related window is minimized, the middleware 520 determines whether the device 100 is in the portrait or landscape display mode and, when in the landscape display mode, which of the touch sensitive displays 110 and 114 is on top and which is on the bottom (step 1916). When in portrait display mode, the middleware 520 causes the lower level window to no longer be displayed on a predetermined one of the left or right touch sensitive display. When in landscape display mode, the middleware 520 causes the lower level window to no longer be displayed on a predetermined one of the upper or lower touch sensitive display.

[0135] The exemplary systems and methods of this disclosure have been described in relation to communication devices. However, to avoid unnecessarily obscuring the present disclosure, the preceding description omits a number of known structures and devices. This omission is not to be construed as a limitation of the scopes of the claims. Specific details are set forth to provide an understanding of the present disclosure. It should however be appreciated that the present disclosure may be practiced in a variety of ways beyond the specific detail set forth herein.

[0136] Furthermore, while the exemplary aspects, embodiments, and/or configurations illustrated herein show the various components of the system collocated, certain components of the system can be located remotely, at distant portions of a distributed network, such as a LAN and/or the Internet, or within a dedicated system. Thus, it should be appreciated, that the components of the system can be combined in to one or more devices, such as a communication device, or collocated on a particular node of a distributed network, such as an analog and/or

digital telecommunications network, a packet-switch network, or a circuit-switched network. It will be appreciated from the preceding description, and for reasons of computational efficiency, that the components of the system can be arranged at any location within a distributed network of components without affecting the operation of the system. For example, the various components can be located in a switch such as a PBX and media server, gateway, in one or more communications devices, at one or more users' premises, or some combination thereof. Similarly, one or more functional portions of the system could be distributed between a telecommunications device(s) and an associated computing device.

[0137] Furthermore, it should be appreciated that the various links connecting the elements can be wired or wireless links, or any combination thereof, or any other known or later developed element(s) that is capable of supplying and/or communicating data to and from the connected elements. These wired or wireless links can also be secure links and may be capable of communicating encrypted information. Transmission media used as links, for example, can be any suitable carrier for electrical signals, including coaxial cables, copper wire and fiber optics, and may take the form of acoustic or light waves, such as those generated during radio-wave and infra-red data communications.

[0138] Also, while the flowcharts have been discussed and illustrated in relation to a particular sequence of events, it should be appreciated that changes, additions, and omissions to this sequence can occur without materially affecting the operation of the disclosed embodiments, configuration, and aspects.

[0139] A number of variations and modifications of the disclosure can be used. It would be possible to provide for some features of the disclosure without providing others.

[0140] In other embodiments, other rules may be applied. For example, the preferred orientations of the higher and lower level related windows is different from the orientations described above.

[0141] In yet another embodiment, the systems and methods of this disclosure can be implemented in conjunction with a special purpose computer, a programmed microprocessor or microcontroller and peripheral integrated circuit element(s), an ASIC or other integrated circuit, a digital signal processor, a hard-wired electronic or logic circuit such as discrete element circuit, a programmable logic device or gate array such as PLD, PLA, FPGA, PAL, special purpose computer, any comparable means, or the like. In general, any device(s) or means capable of

implementing the methodology illustrated herein can be used to implement the various aspects of this disclosure. Exemplary hardware that can be used for the disclosed embodiments, configurations and aspects includes computers, handheld devices, telephones (e.g., cellular, Internet enabled, digital, analog, hybrids, and others), and other hardware known in the art. Some of these devices include processors (e.g., a single or multiple microprocessors), memory, nonvolatile storage, input devices, and output devices. Furthermore, alternative software implementations including, but not limited to, distributed processing or component/object distributed processing, parallel processing, or virtual machine processing can also be constructed to implement the methods described herein.

[0142] In yet another embodiment, the disclosed methods may be readily implemented in conjunction with software using object or object-oriented software development environments that provide portable source code that can be used on a variety of computer or workstation platforms. Alternatively, the disclosed system may be implemented partially or fully in hardware using standard logic circuits or VLSI design. Whether software or hardware is used to implement the systems in accordance with this disclosure is dependent on the speed and/or efficiency requirements of the system, the particular function, and the particular software or hardware systems or microprocessor or microcomputer systems being utilized.

[0143] In yet another embodiment, the disclosed methods may be partially implemented in software that can be stored on a storage medium, executed on programmed general-purpose computer with the cooperation of a controller and memory, a special purpose computer, a microprocessor, or the like. In these instances, the systems and methods of this disclosure can be implemented as program embedded on personal computer such as an applet, JAVA® or CGI script, as a resource residing on a server or computer workstation, as a routine embedded in a dedicated measurement system, system component, or the like. The system can also be implemented by physically incorporating the system and/or method into a software and/or hardware system.

[0144] Although the present disclosure describes components and functions implemented in the aspects, embodiments, and/or configurations with reference to particular standards and protocols, the aspects, embodiments, and/or configurations are not limited to such standards and protocols. Other similar standards and protocols not mentioned herein are in existence and are considered to be included in the present disclosure. Moreover, the standards and protocols

mentioned herein and other similar standards and protocols not mentioned herein are periodically superseded by faster or more effective equivalents having essentially the same functions. Such replacement standards and protocols having the same functions are considered equivalents included in the present disclosure.

[0145] The present disclosure, in various aspects, embodiments, and/or configurations, includes components, methods, processes, systems and/or apparatus substantially as depicted and described herein, including various aspects, embodiments, configurations embodiments, subcombinations, and/or subsets thereof. Those of skill in the art will understand how to make and use the disclosed aspects, embodiments, and/or configurations after understanding the present disclosure. The present disclosure, in various aspects, embodiments, and/or configurations, includes providing devices and processes in the absence of items not depicted and/or described herein or in various aspects, embodiments, and/or configurations hereof, including in the absence of such items as may have been used in previous devices or processes, e.g., for improving performance, achieving ease and\or reducing cost of implementation.

[0146] The foregoing discussion has been presented for purposes of illustration and description. The foregoing is not intended to limit the disclosure to the form or forms disclosed herein. In the foregoing Detailed Description for example, various features of the disclosure are grouped together in one or more aspects, embodiments, and/or configurations for the purpose of streamlining the disclosure. The features of the aspects, embodiments, and/or configurations of the disclosure may be combined in alternate aspects, embodiments, and/or configurations other than those discussed above. This method of disclosure is not to be interpreted as reflecting an intention that the claims require more features than are expressly recited in each claim. Rather, as the following claims reflect, inventive aspects lie in less than all features of a single foregoing disclosed aspect, embodiment, and/or configuration. Thus, the following claims are hereby incorporated into this Detailed Description, with each claim standing on its own as a separate preferred embodiment of the disclosure.

[0147] Moreover, though the description has included description of one or more aspects, embodiments, and/or configurations and certain variations and modifications, other variations, combinations, and modifications are within the scope of the disclosure, e.g., as may be within the skill and knowledge of those in the art, after understanding the present disclosure. It is intended to obtain rights which include alternative aspects, embodiments, and/or configurations to the

extent permitted, including alternate, interchangeable and/or equivalent structures, functions, ranges or steps to those claimed, whether or not such alternate, interchangeable and/or equivalent structures, functions, ranges or steps are disclosed herein, and without intending to publicly dedicate any patentable subject matter.

What is claimed is:

1. A method, comprising:

receiving, by processor executable middleware in a multi-display communication device, a command to maximize a set of related windows ftom a common multi-screen application, the set of related windows comprising a higher level window and a first lower level window at least one of displayed and to be displayed by different touch sensitive displays;

selecting, by the processor executable middleware, a predetermined touch sensitive display for displaying one of the higher and first lower level window and causing the other of the higher and first lower level window to be displayed on the other touch sensitive display;

receiving, by the processor executable middleware, a selection of an item displayed in the higher level window; and

in response, opening and displaying, by the processor executable middleware and in the touch sensitive display displaying the first lower level window, a second lower level window comprising the selected item.

2. The method of claim 1, wherein the common multi-screen application is an email application, wherein the higher level view is an inbox, and wherein the selected item is an email listed in the inbox.

3. The method of claim 1, wherein the command is a gesture received by an off-display gesture capture region and the selection is a gesture received by the touch sensitive display displaying the higher level window.

4. The method of claim 1, wherein, when the second lower level window is opened and displayed, the first lower level window is closed automatically.

5. The method of claim 1, wherein the command is a drag or a flick, wherein the lower level window is in the predetermined touch sensitive display when the gesture is received, wherein the higher level window appears to move from the predetermined touch sensitive display to the other touch sensitive display, whereby the lower level window is displayed by the predetermined touch sensitive display and the higher level window is displayed by the other touch sensitive display.

6. The method of claim 2, wherein the multi-display communication device is in the portrait display mode and wherein the predetermined touch sensitive display is the right-most touch sensitive display.

7. A non-transient and tangible computer readable medium comprising processor executable instructions to perform the steps of claim 1.

8. A communication device, comprising:

at least first and second touch sensitive displays to display windows;

a processor; and

a computer readable memory, the computer readable memory comprising processor executable middleware operable to:

receive a command to maximize a set of related windows ftom a common multi-screen application, the set of related windows comprising a higher level window and a first lower level window at least one of displayed and to be displayed by different touch sensitive displays;

select a predetermined touch sensitive display for displaying one of the higher and first lower level window and causing the other of the higher and first lower level window to be displayed on the other touch sensitive display;

receive a selection of an item displayed in the higher level window; and

in response, open and display, in the touch sensitive display displaying the first lower level window, a second lower level window comprising the selected item.

9. The device of claim 8, wherein the common multi-screen application is an email application, wherein the higher level view is an inbox, and wherein the selected item is an email listed in the inbox.

10. The device of claim 8, wherein the command is a gesture received by an off-display gesture capture region and the selection is a gesture received by the touch sensitive display displaying the higher level window.

11. The device of claim 8, wherein, when the second lower level window is opened and displayed, the first lower level window is closed automatically.

12. The device of claim 8, wherein the command is a drag or a flick, wherein the lower level window is in the predetermined touch sensitive display when the gesture is received, wherein the higher level window appears to move from the predetermined touch sensitive display to the other touch sensitive display, whereby the lower level window is displayed by the predetermined touch sensitive display and the higher level window is displayed by the other touch sensitive display.

13. The device of claim 8, wherein the multi-display communication device is in the portrait display mode and wherein the predetermined touch sensitive display is the right-most touch sensitive display.

14. A method, comprising:

receiving, by an off-display gesture region in a multi-display communication device, a command to one of maximize and minimize a set of related windows; and

in response, performing, by processor executable middleware, the one of maximizing and minimizing the set of related windows.

15. The method of claim 14, wherein the one of maximizing and minimizing is maximizing, wherein the set of related windows comprises higher and lower level hierarchically related windows and further comprising:

selecting a predetermined touch sensitive display for displaying one of the higher and lower level windows and causing the other of the higher and lower level windows to be displayed on the other touch sensitive display.

16. The method of claim 15, wherein the command is a drag or a flick, wherein the higher and lower level window are concurrently displayed in the selected and other touch sensitive displays when the gesture is received, wherein the lower level display is displayed in the selected touch sensitive display when the gesture is received, and wherein the higher level window appears to move from the other touch sensitive display to the selected touch sensitive display, thereby covering the lower level window in the other touch sensitive display.

17. The method of claim 15, wherein the command is a drag or a flick, wherein the higher and lower level window are concurrently displayed in the selected and other touch sensitive displays when the gesture is received, wherein the higher level display is displayed in the selected touch sensitive display when the gesture is received, and wherein the lower level window appears to move from the other touch sensitive display to the selected touch sensitive display, thereby disappearing under the higher level window in the other touch sensitive display.

18. The method of claim 14, wherein the one of maximizing and minimizing is minimizing, wherein the set of related windows comprises higher and lower level hierarchically related windows and further comprising:

selecting a touch sensitive display for displaying the higher level window and ceasing to display the lower level window on the other touch sensitive display.

19. The method of claim 18, wherein the command is a drag or a flick, wherein neither the higher nor lower level window is displayed in the predetermined touch sensitive display when the gesture is received, and wherein the lower level window appears to move from the other touch sensitive display to the predetermined touch sensitive display, thereby revealing the higher level window in the other touch sensitive display.

20. The method of claim 18, wherein the command is a drag or a flick, wherein the lower level window is in the predetermined touch sensitive display when the gesture is received, wherein the higher level window appears to move from the predetermined touch sensitive display to the other touch sensitive display, whereby the lower level window is displayed by the predetermined touch sensitive display and the higher level window is displayed by the other touch sensitive display.

21. The method of claim 14, wherein a parent and child window in the set of related windows are concurrently displayed on differing touch sensitive displays and are nonoverlapping.

22. A non-transient and tangible computer readable medium comprising processor executable instructions to perform the steps of claim 14.

23. A multi-display communication device operable to perform the steps of claim 14.

ABSTRACT

[0148] The present disclosure is directed to methodologies and devices for handling maximizing and minimizing of hierarchically related windows.

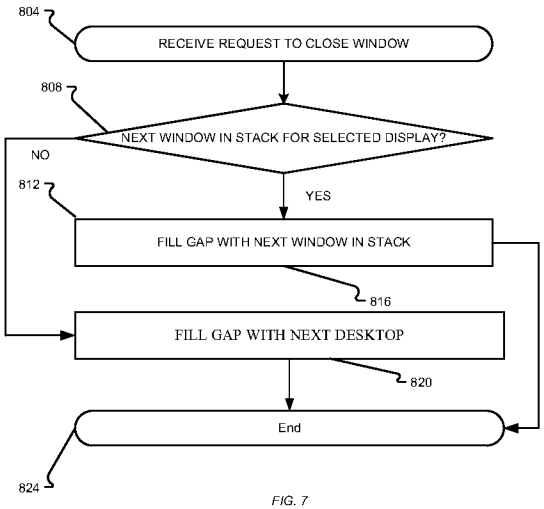


FIG. 7

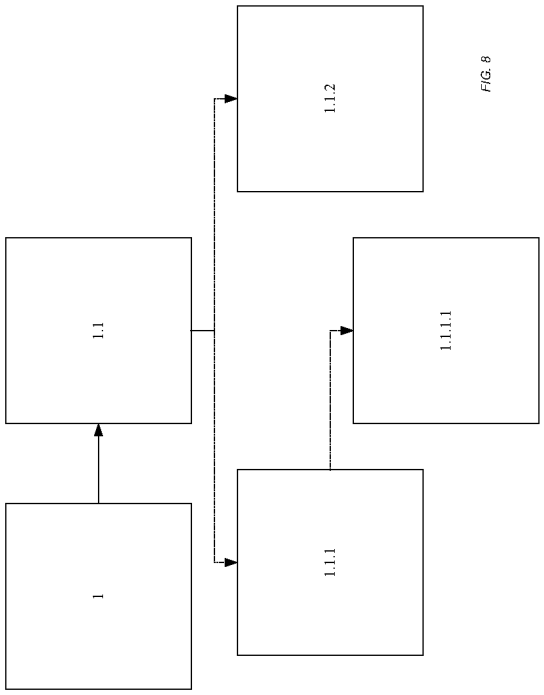


FIG. 8

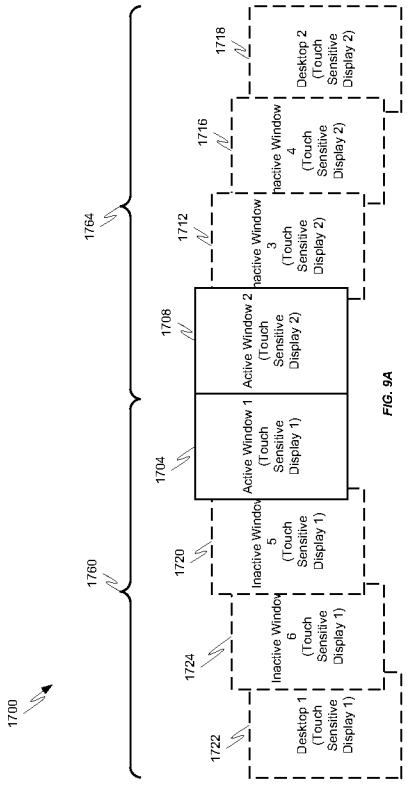


FIG. 9A

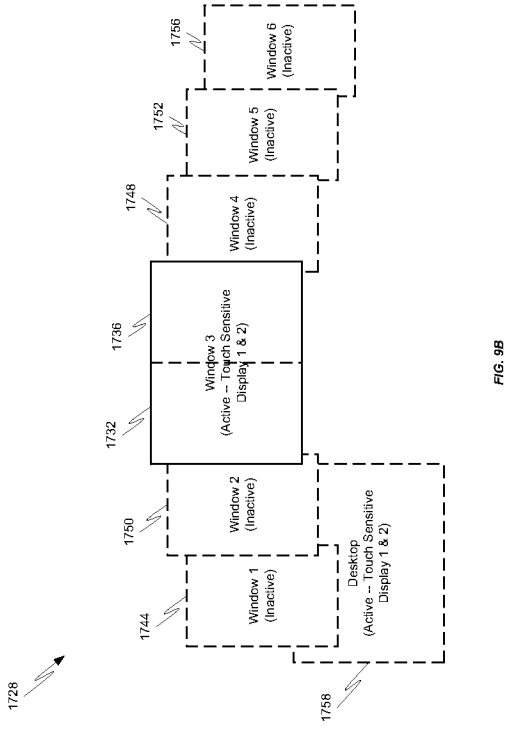


FIG. 9B

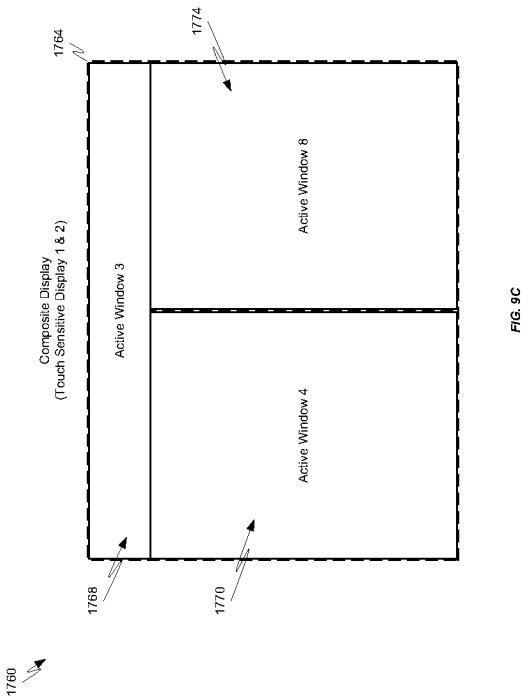


FIG. 9C

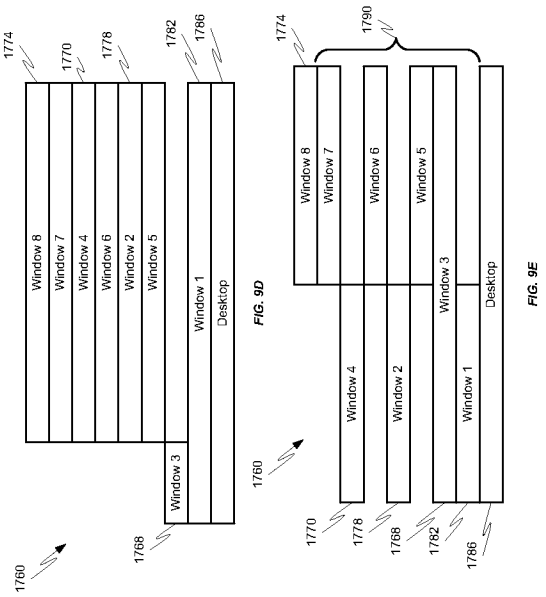


FIG. 9D

FIG. 9E

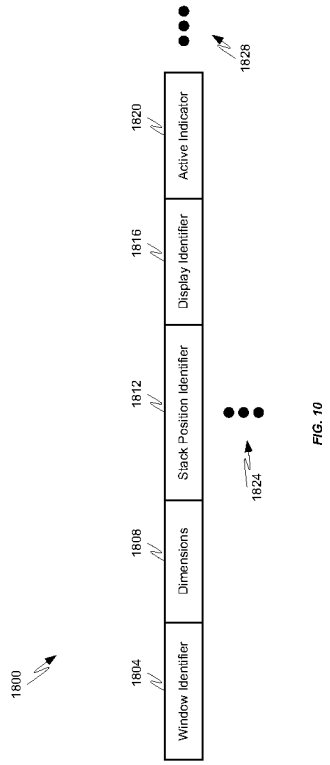


FIG. 10

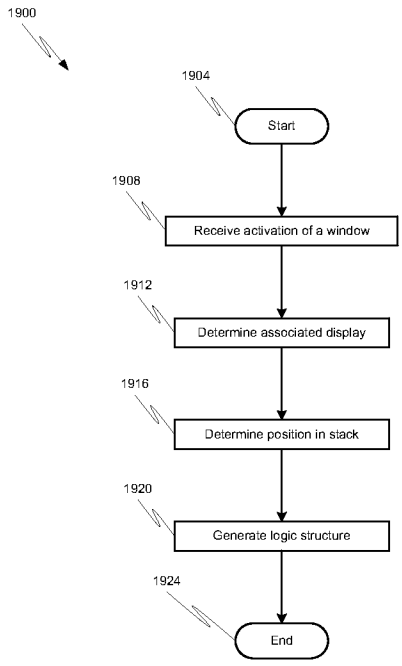


FIG. 11

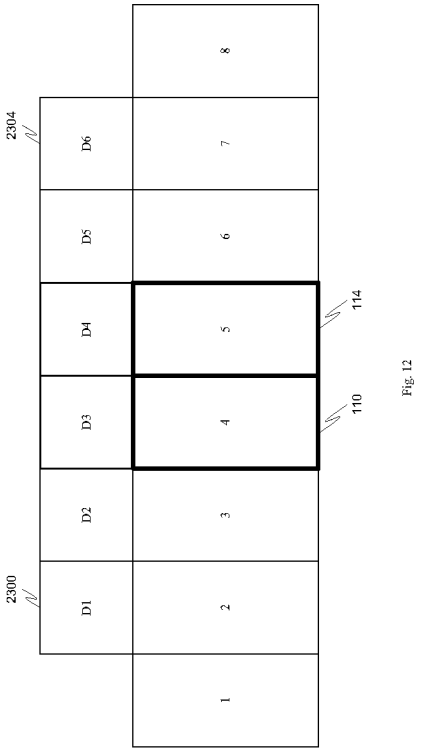


Fig. 12

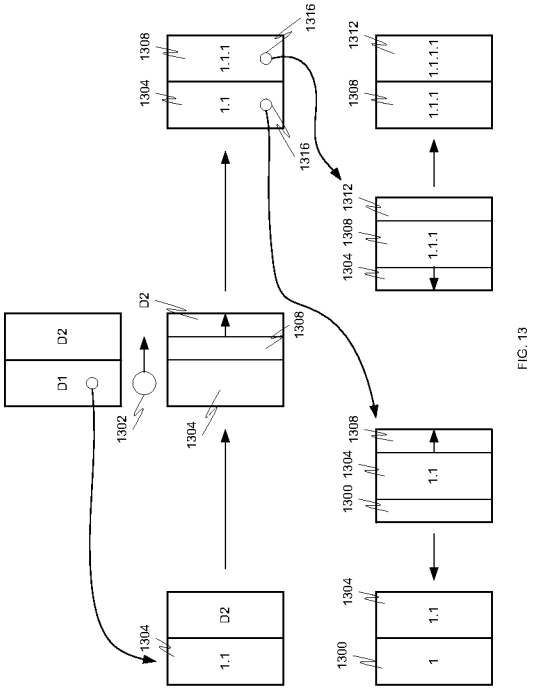


FIG. 13

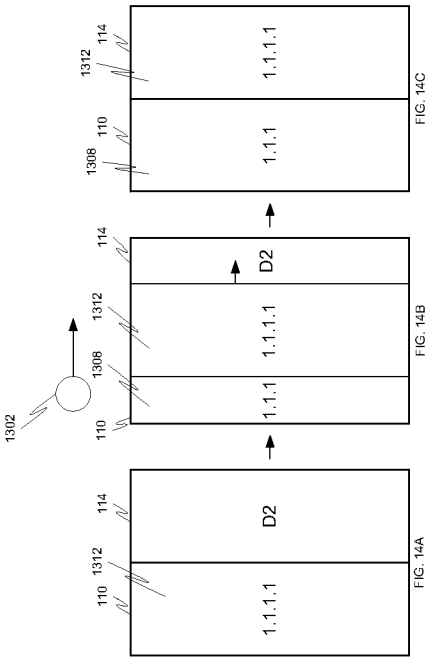


FIG. 14A

FIG. 14B

FIG. 14C

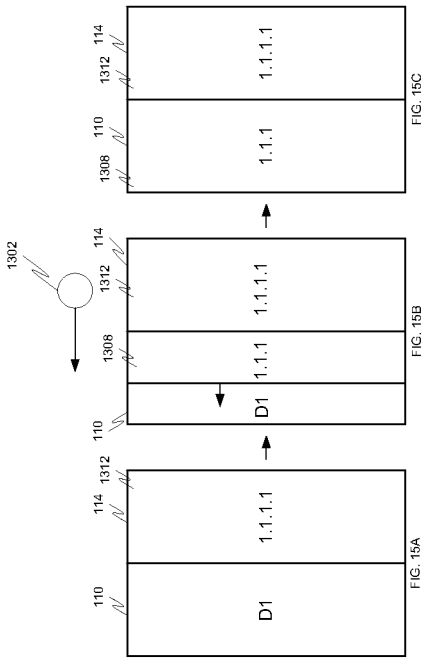
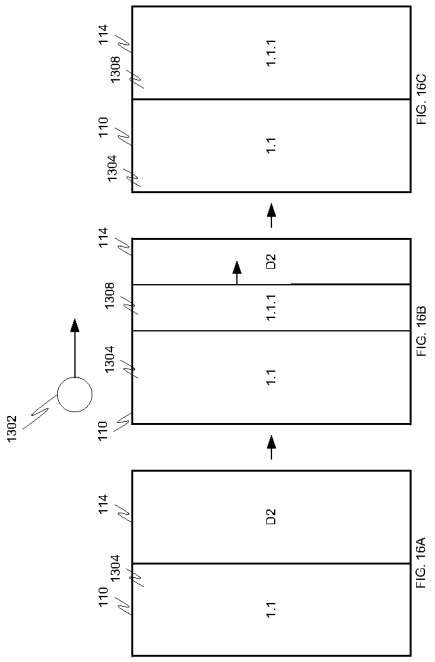
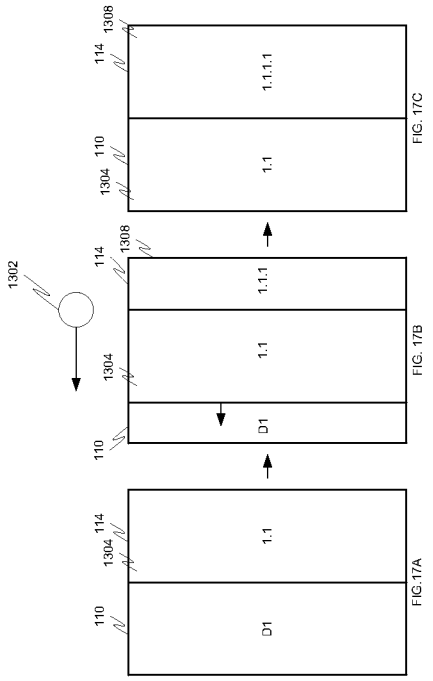


FIG. 15A

FIG. 15B

FIG. 15C


1.1.1
1.1
D2
1.1.1
1.1
D2
1.1
1302
114
1308
1304
110
FIG. 16C
FIG. 16B
FIG. 16A


1.1.1.1
1.1
1.1.1
1.1
D1
1.1
D1
1302
114
1308
1304
110
FIG. 17C
FIG. 17B
FIG.17A

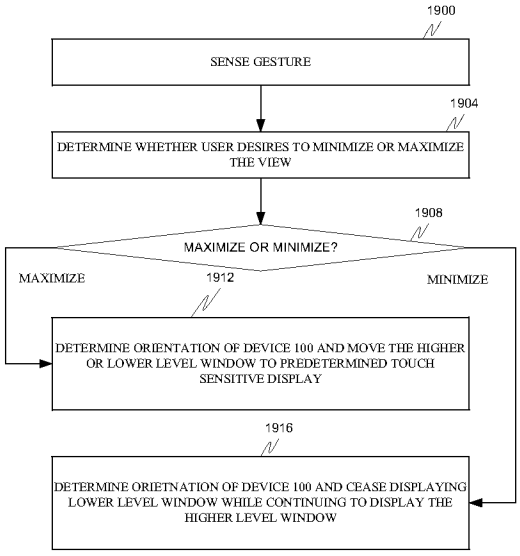

1900
SENSE GESTURE
1904
DETERMINE WHETHER USER DESIRES TO MINIMIZE OR MAXIMIZE THE VIEW
1908
MAXIMIZE OR MINIMIZE?
MAXIMIZE
MINIMIZE
1912
DETERMINE ORIENTATION OF DEVICE 100 AND MOVE THE HIGHER OR LOWER LEVEL WINDOW TO PREDETERMINED TOUCH SENSITIVE DISPLAY
1916
DETERMINE ORIETNATION OF DEVICE 100 AND CEASE DISPLAYING LOWER LEVEL WINDOW WHILE CONTINUING TO DISPLAY THE HIGHER LEVEL WINDOW

FIG. 19

MANAGING HIERARCHICALLY RELATED WINDOWS IN A SINGLE DISPLAY

[0100] The device 100 manages desktops and/or windows with at least one window stack 1700, 1728, as shown in Figs. 9A and 9B. A window stack 1700, 1728 is a logical arrangement of active and/or inactive windows for a multi-screen device. For example, the window stack 1700, 1728 may be logically similar to a deck of cards, where one or more windows or desktops are arranged in order, as shown in Figs. 9A and 9B. An active window is a window that is currently being displayed on at least one of the touch sensitive displays 110, 114. For example, windows 104 and 108 are active windows and are displayed on touch sensitive displays 110 and 114. An inactive window is a window that was opened and displayed but is now "behind" an active window and not being displayed. In embodiments, an inactive window may be for an application that is suspended, and thus, the window is not displaying active content. For example, windows 1712, 1716, 1720, and 1724 are inactive windows.

[0101] A window stack 1700, 1728 may have various arrangements or organizational structures. In the embodiment shown in Fig. 9A, the device 100 includes a first stack 1760 associated with a first touch sensitive display 110 and a second stack associated with a second touch sensitive display 114. Thus, each touch sensitive display 110, 114 can have an associated window stack 1760, 1764. These two window stacks 1760, 1764 may have different numbers of windows arranged in the respective stacks 1760, 1764. Further, the two window stacks 1760, 1764 can also be identified differently and managed separately. Thus, the first window stack 1760 can be arranged in order from a first window 1704 to a next window 1720 to a last window 1724 and finally to a desktop 1722, which, in embodiments, is at the "bottom" of the window stack 1760. In embodiments, the desktop 1722 is not always at the "bottom" as application windows can be arranged in the window stack below the desktop 1722, and the desktop 1722 can be brought to the "top" of a stack over other windows during a desktop reveal. Likewise, the second stack 1764 can be arranged from a first window 1708 to a next window 1712 to a last window 1716, and finally to a desktop 1718, which, in embodiments, is a single desktop area, with desktop 1722, under all the windows in both window stack 1760 and window stack 1764. A logical data structure for managing the two window stacks 1760, 1764 may be as described in conjunction with Fig. 10.

[0102] Another arrangement for a window stack 1728 is shown in Fig. 9B. In this embodiment, there is a single window stack 1728 for both touch sensitive displays 110, 114.

a function of the orientation of the device 100, the context of what programs, functions, software, etc. are being executed on the device 100, how the stack is positioned when the new window is opened, etc. To insert the window in the stack, the position in the stack for the window is determined and the touch sensitive display 110, 114 to which the window is associated may also be determined. With this information, a logical data structure for the window can be created and stored. When user interface or other events or tasks change the arrangement of windows, the window stack(s) can be changed to reflect the change in arrangement. It should be noted that these same concepts described above can be used to manage the one or more desktops for the device 100.

[0106] A logical data structure 1800 for managing the arrangement of windows or desktops in a window stack is shown in Fig. 10. The logical data structure 1800 can be any data structure used to store data whether an object, record, file, etc. The logical data structure 1800 can be stored in any type of database or data storage system, regardless of protocol or standard. In embodiments, the logical data structure 1800 includes one or more portions, fields, attributes, etc. that store data in a logical arrangement that allows for easy storage and retrieval of the information. Hereinafter, these one or more portions, fields, attributes, etc. shall be described simply as fields. The fields can store data for a window identifier 1804, dimensions 1808, a stack position identifier 1812, a display identifier 1816, and/or an active indicator 1820. Each window in a window stack can have an associated logical data structure 1800. While only a single logical data structure 1800 is shown in Fig. 10, there may be more or fewer logical data structures 1800 used with a window stack (based on the number of windows or desktops in the stack), as represented by ellipses 1824. Further, there may be more or fewer fields than those shown in Fig. 10, as represented by ellipses 1828.

[0107] A window identifier 1804 can include any identifier (ID) that uniquely identifies the associated window in relation to other windows in the window stack. The window identifier 1804 can be a globally unique identifier (GUID), a numeric ID, an alphanumeric ID, or other type of identifier. In embodiments, the window identifier 1804 can be one, two, or any number of digits based on the number of windows that can be opened. In alternative embodiments, the size of the window identifier 1804 may change based on the number of windows opened. While the window is open, the window identifier 1804 may be static and remain unchanged.

Thus, the window stack 1728 is arranged from a desktop 1758 to a first window 1744 to a last window 1756. A window can be arranged in a position among all windows without an association to a specific touch sensitive display 110, 114. In this embodiment, a window is in the order of windows. Further, at least one window is identified as being active. For example, a single window may be rendered in two portions 1732 and 1736 that are displayed on the first touch sensitive screen 110 and the second touch sensitive screen 114. The single window may only occupy a single position in the window stack 1728 although it is displayed on both displays 110, 114.

[0103] Yet another arrangement of a window stack 1760 is shown in Figs. 9C through 9E. The window stack 1760 is shown in three "elevation" views. In Fig. 9C, the top of the window stack 1760 is shown. Two sides of the window stack 1760 are shown in Figs. 9D and 9E. In this embodiment, the window stack 1760 resembles a stack of bricks. The windows are stacked on each other. Looking from the top of the window stack 1760 in Fig, 9C, only the top most windows in the window stack 1760 are seen in different portions of the composite display 1764. The composite display 1764 represents a logical model for the entire display area of the device 100, which can include touch sensitive display 110 and touch sensitive display 114. A desktop 1786 or a window can occupy part or all of the composite display 1764.

[0104] In the embodiment shown, the desktop 1786 is the lowest display or "brick" in the window stack 1760. Thereupon, window 1 1782, window 2 1782, window 3 1768, and window 4 1770 are layered. Window 1 1782, window 3 1768, window 2 1782, and window 4 1770 only occupy a portion of the composite display 1764. Thus, another part of the stack 1760 includes window 8 1774 and windows 5 through 7 shown in section 1790. Only the top window in any portion of the composite display 1764 is actually rendered and displayed. Thus, as shown in the top view in Fig. 9C, window 4 1770, window 8 1774, and window 3 1768 are displayed as being at the top of the display in different portions of the window stack 1760. A window can be dimensioned to occupy only a portion of the composite display 1760 to "reveal" windows lower in the window stack 1760. For example, window 3 1768 is lower in the stack than both window 4 1770 and window 8 1774 but is still displayed. A logical data structure to manage the window stack can be as described in conjunction with Fig. 10.

[0105] When a new window is opened, the newly activated window is generally positioned at the top of the stack. However, where and how the window is positioned within the stack can be

[0108] Dimensions 1808 can include dimensions for a window in the composite display 1760. For example, the dimensions 1808 can include coordinates for two or more corners of the window or may include one coordinate and dimensions for the width and height of the window. These dimensions 1808 can delineate what portion of the composite display 1760 the window may occupy, which may the entire composite display 1760 or only part of composite display 1760. For example, window 4 1770 may have dimensions 1880 that indicate that the window 1770 will occupy only part of the display area for composite display 1760, as shown in Figs. 9C through 9E. As windows are moved or inserted in the window stack, the dimensions 1808 may change.

[0109] A stack position identifier 1812 can be any identifier that can identify the position in the stack for the window or may be inferred from the window's control record within a data structure, such as a list or a stack. The stack position identifier 1812 can be a GUID, a numeric ID, an alphanumeric ID, or other type of identifier. Each window or desktop can include a stack position identifier 1812. For example, as shown in Fig. 9A, window 1 1704 in stack 1 1760 can have a stack position identifier 1812 of 1 identifying that window 1704 is the first window in the stack 1760 and the active window. Similarly, window 6 1724 can have a stack position identifier 1812 of 3 representing that window 1724 is the third window in the stack 1760. Window 2 1708 can also have a stack position identifier 1812 of 1 representing that window 1708 is the first window in the second stack 1764. As shown in Fig. 9B, window 1 1744 can have a stack position identifier 1812 of 1, window 3, rendered in portions 1732 and 1736, can have a stack position identifier 1812 of 3, and window 6 1756 can have a stack position identifier 1812 of 6. Thus, depending on the type of stack, the stack position identifier 1812 can represent a window's location in the stack.

[0110] A display identifier 1816 can identify that the window or desktop is associated with a particular display, such as the first display 110 or the second display 114, or the composite display 1760 composed of both displays. While this display identifier 1816 may not be needed for a multi-stack system, as shown in Fig. 9A, the display identifier 1816 can indicate whether a window in the serial stack of Fig. 9B is displayed on a particular display. Thus, window 3 may have two portions 1732 and 1736 in Fig. 9B. The first portion 1732 may have a display identifier 1816 for the first display while the second portion 1736 may have a display identifier 1816 for the second display 114. However, in alternative embodiments, the window may have

two display identifier 1816 that represent that the window is displayed on both of the displays 110, 114, or a display identifier 1816 identifying the composite display. In another alternate embodiment, the window may have a single display identifier 1816 to represent that the window is displayed on both of the displays 110, 114.

[0111] Similar to the display identifier 1816, an active indicator 1820 may not be needed with the dual stack system of Fig. 9A, as the window in stack position 1 is active and displayed. In the system of Fig. 9B, the active indicator 1820 can indicate which window(s) in the stack is being displayed. Thus, window 3 may have two portions 1732 and 1736 in Fig. 9. The first portion 1732 may have an active indicator 1820 while the second portion 1736 may also have an active indicator 1820. However, in alternative embodiments, window 3 may have a single active indicator 1820. The active indicator 1820 can be a simple flag or bit that represents that the window is active or displayed.

[0112] An embodiment of a method 900 for creating a window stack is shown in Fig. 11. While a general order for the steps of the method 1900 is shown in Fig. 11. Generally, the method 1900 starts with a start operation 1904 and ends with an end operation 1928. The method 1900 can include more or fewer steps or can arrange the order of the steps differently than those shown in Fig. 11. The method 1900 can be executed as a set of computer-executable instructions executed by a computer system and encoded or stored on a computer readable medium. Hereinafrer, the method 1900 shall be explained with reference to the systems, components, modules, software, data structures, user interfaces, etc. described in conjunction with Figs. 1-10.

[0113] A multi-screen device 100 can receive activation of a window, in step 1908. In embodiments, the multi-screen device 100 can receive activation of a window by receiving an input from the touch sensitive display 110 or 114, the configurable area 112 or 116, a gesture capture region 120 or 124, or some other hardware sensor operable to receive user interface inputs. The processor may execute the Task Management Module 540 may receive the input. The Task Management Module 540 can interpret the input as requesting an application task to be executed that will open a window in the window stack.

[0114] In embodiments, the Task Management Module 540 places the user interface interaction in the task stack 552 to be acted upon by the Display Configuration Module 568 of the Multi-Display Management Module 524. Further, the Task Management Module 540 waits

for information from the Multi-Display Management Module 524 to send instructions to the Window Management Module 532 to create the window in the window stack.

[0115] The Multi-Display Management Module 524, upon receiving instruction from the Task Management Module 540, determines to which touch portion of the composite display 1760, the newly activated window should be associated, in step 1912. For example, window 4 1770 is associated with the a portion of the composite display 1764 In embodiments, the device state module 574 of the Multi-Display Management Module 524 may determine how the device is oriented or in what state the device is in, e.g., open, closed, portrait, etc. Further, the preferences module 572 and/or requirements module 580 may determine how the window is to be displayed. The gesture module 576 may determine the user's intentions about how the window is to be opened based on the type of gesture and the location of where the gesture is made.

[0116] The Display Configuration Module 568 may use the input from these modules and evaluate the current window stack 1760 to determine the best place and the best dimensions, based on a visibility algorithm, to open the window. Thus, the Display Configuration Module 568 determines the best place to put the window at the top of the window stack 1760, in step 1916. The visibility algorithm, in embodiments, determines for all portions of the composite display, which windows are at the top of the stack. For example, the visibility algorithm determines that window 3 1768, window 4 1770, and window 8 1774 are at the top of the stack 1760 as viewed in Figs. 9C through 9E. Upon determining where to open the window, the Display Configuration Module 568 can assign a display identifier 816 and possibly dimensions 808 to the window. The display identifier 816 and dimensions 808 can then be sent back to the Task Management Module 540. The Task Management Module 540 may then assign the window a stack position identifier 812 indicating the windows position at the top of the window stack.

[0117] In embodiments, the Task Management Module 540 sends the window stack information and instructions to render the window to the Window Management Module 532. The Window Management Module 532 and the Task Management Module 540 can create the logical data structure 800, in step 1924. Both the Task Management Module 540 and the Window Management Module 532 may create and manage copies of the window stack. These copies of the window stack can be synchronized or kept similar through communications between the Window Management Module 532 and the Task Management Module 540. Thus,

the Window Management Module 532 and the Task Management Module 540, based on the information determined by the Multi-Display Management Module 524, can assign dimensions 808, a stack position identifier 812 (e.g., window 1 1782, window 4 1770, etc.), a display identifier 816 (e.g., touch sensitive display 1 110, touch sensitive display 2 114, composite display identifier, etc,), and an active indicator 820, which is generally always set when the window is at the "top" of the stack. The logical data structure 800 may then be stored by both the Window Management Module 532 and the Task Management Module 540. Further, the Window Management Module 532 and the Task Management Module 540may thereinafrer manage the window stack and the logical data structure(s) 800.

[0118] FIG. 12 depicts a further window stacking configuration. A plurality of windows 1, 2, 3, 4, 5, 6, 7, and 8, whether from the same or different multi-screen or single-screen applications, are depicted. The touch sensitive display 110 currently has window 4 in the active display position while touch sensitive display 114 currently has window 5 in the active display position. The stack for touch sensitive display 110, from top to bottom, has window 4 in the active display position and windows 3, 2, and 1 positioned behind it. The stack for touch sensitive display 114, from top to bottom, has window 5 in the active display position and windows 6, 7, and 8 positioned behind it.

[0119] The desktops D1, D2, D3, D4, D5, and D6 are positioned behind the window stacks. The desktops can be viewed as a desktop stack different from the window stack. Viewed in this manner, the touch sensitive display 110 has a corresponding desktop stack comprising desktops D3, D2, and D1,with desktop D1 being at the bottom 2300 stack position and with desktop D3 at the top stack position being capable of being displayed with window 4 (depending on the window position and size (whether maximized or minimized), and the touch sensitive display 114 has a corresponding desktop stack has a corresponding desktop stack comprising desktops D4, D5, and D6,with desktop D6 being at the bottom 2304 stack position and with desktop D4 at the top stack position being capable of being displayed with window 5 (depending on the window position and size (whether maximized or minimized). Conceptually, the desktops can be viewed as a canvas divided, in this example, into six sections of which two can be displayed, at any one time, the touch sensitive displays 110, 114. When the device 100 is in the closed state, this conceptual model, in one configuration, persists. In this configuration, only one window and desktop stacks can be seen (that corresponding to the primary screen) but the other

window and desktop stacks are virtual; that is, they are maintained in memory but cannot be seen because the secondary screen is not enabled.

[0120] Single-screen applications are typically related only to one window while multi-screen applications can be related to one or more windows. The latter hierarchical relationship is depicted in Fig. 8. As shown in Fig. 8, first window 1 is a root window, second and third windows 1.1 and 1.1.1 are node windows, and fourth and fifth windows 1.1.1.1 and 1.1.2 are leaf windows. The windows are in a hierarchical tree-like sequence and are controlled by a common multi-screen application. The root first window 1 may be a top level view of the hierarchical application window sequence such that there is no parent window corresponding to the root window. The root first window 1 may be a parent to a node window, such as second window 1.1 and the second window 1.1 is a child of the first window 1. One or more node windows may be provided that are related as parent/children. A node window may also serve as a leaf window. By leaf window, it is meant that the leaf window has no corresponding node screen for which the leaf window is a parent (or a leaf window has no child windows). As such, the leaf window does not have any children node windows, Fourth and fifth windows 1.1.1.1 and 1.1.2 are examples of leaf windows. The root view refers to all of the windows controlled by a multi-screen application at a point in time.

[0121] A number of examples will now be discussed with reference to minimizing and maximizing root view windows and subcomponents thereof. Prior to discussing the examples, some fundamental rules regarding maximizing and minimizing displayed images will be discussed. A maximize or minimize operation is typically initiated by a user gesture, particularly a flip or drag. In a drag gesture, a 1:1 map reaction between the user's digit and the displayed image is used as the user "drags" the displayed image to the other touch sensitive display. The user typically releases the displayed image before it is fully moved to the other touch sensitive display. The initial speed after release is generally decreased progressively or gradually until the displayed image is fully located in the desired position. Too little displacement of the displayed image generally will not cause the image to be relocated. Such a low level of displacement is assumed to be accidental or incidental to the user using the device.

[0122] In a root view, or set of hierarchically related windows, the user, when the device is in dual screen mode, can maximize the related windows of the multi-screen application to enable side-by-side viewing of the related windows (e.g., the parent and child windows), thereby

viewing the window hierarchy as a linear progression, with the top-most window being displayed on the left and the lowest-most or deepest level window being displayed on the right (or vice versa). As discussed below, this visual configuration can be controlled using directional transitions conveying a navigation direction and location of a window in relation to a launching point of the window.

[0123] A multi-screen application window generally opens in minimized mode. A continuing drag and release gesture when a window of a multi-screen application, originally minimized in a first touch sensitive display, is at least half-way across the second touch sensitive display will generally cause the window to be maximized, or displayed in both the first and second touch sensitive displays. Dragging a window of a multi-screen application originally minimized in a first touch sensitive display to a second touch sensitive display all the way across to the far edge of the second touch sensitive display will generally maximize the window across both the first and second touch sensitive displays. When the user continues to drag in the same direction after the window of the multi-screen application is expanded fully (or maximized across both the first and second touch sensitive displays and touching both far (non-adjacent) edges of the first and second touch sensitive displays), the window will pull away from the far edge of the first touch sensitive display and, once the far edge of the window is half-way across the first touch sensitive display, the application window will generally reposition in minimized mode in the second touch sensitive display.

[0124] The flick gesture will not generally maximize a window of a multi-screen application but generally will reposition a minimized window of a single- or multi-screen application. The flick gesture will generally not minimize window(s) of a maximized application. The flick gesture is a way to move rapidly through the windows of a stack, whether the windows are of a single- or multi-screen application.

[0125] A first example will be discussed with reference to Figs. 13A-C. The communication device 100 is in a single-screen mode in which the second touch sensitive display 114 is inactive, such as is the case when the communication device 100 is in the closed state or certain state transition modes. In this mode, the parent/child hierarchy is not apparent. The parent window launches a child window, and the user needs to select back to redisplay the parent window so that the user can launch another child window. A user selects an application shortcut on a first desktop D1 displayed by the first touch sensitive display 110 to launch a multi-screen

application, such as Outlook™. The launching can be, for example, initiated by a tap gesture 1316 on the shortcut. The application opens to display a defined parent or node second window 1.1 (1304) in the first touch sensitive display 110 (Fig. 13A).

[0126] The user then selects an icon, link, or item on the second window 1.1 (1304), such as by a tap gesture 1316 on the selected item, to open a child window, such as a node third window 1.1.1(1308). As can be seen in Fig. 13B, the node third window 1.1.1 (1308) moves over the first touch sensitive display 110 from a first side 1354 while the second window 1.1 (1304) moves out of the first touch sensitive display 110 by exiting the opposing second side 1350 of the first touch sensitive display 110. The third window 1.1.1 can, for example, be an email or set of emails, a contact or set of contacts, a voice mail or set of voice mails, a message or set of messages, and the like. Fig. 13C depicts the final appearance of the first touch sensitive display 110 in which the third window 1.1.1 (1308) is displayed.

[0127] In Fig. 14A, the third window 1.1.1 (1308) is displayed. To get back to the second window 1.1 (1304), the user, by a flick gesture 1400 received by the off-display gesture capture region 120 towards the first side 1354, causes the third window 1.1.1 (1308) to exit the first side 1354 of the first touch sensitive display 110 and the second window 1.1 (1304) to enter from the second side 1350 of the first touch sensitive display 110 (Fig. 14B). Fig. 14C depicts the final appearance of the first touch sensitive display 110 in which the second window 1.1 (1304) is again displayed. Alternatively, the user press a "back" button or arrow to navigate back to the second window 1.1 (1304).

[0128] In Fig. 15A, the third window 1.1.1 (1308) is displayed. The user now desires to a window of a different application in the window stack. He or she therefore, by a drag gesture 1504 in the off-display gesture capture region 120 moves the third window 1.1.1 (1308) towards the first side 1354, where the third window 1.1.1 (1308) exits. At the same time, the window 1500 of the second different application enters from the second side 1350 of the first touch sensitive display 110. Fig. 13C depicts the final appearance of the first touch sensitive display 110 in which the window 1500 of the second application is displayed.

[0129] As can be seen from the above examples, a higher level (or parent) window slides into the first touch sensitive display 110 from the second side 1350 while a lower level (or child) window slides into the first touch sensitive display 110 from the first side 1354.

[0130] In the various examples, middleware 520, particularly one or more of the Multi-Display Management (MDM) class 524, a Surface Cache class 528, a Window Management class 532, an Activity Management class 536, and an Application Management class 540, independently or collectively, receive a command to manipulate a displayed set of related windows from a multi-screen application in a predetermined manner, such as by detecting a gesture (step 1600). In response, middleware 520 determines that the communication device 100 is currently in single-screen mode (step 1604). The middleware 520 next determines whether the selected window to be displayed by the gesture is a parent or child window (decision diamond 1608). When the selected window is a parent window, the middleware 520 causes the parent window to enter the touch sensitive display 110 from the second side 1354 and the child window 1350 to exit through the first side 1350 (step 1612). When the selected window is a child window, the middleware 520 causes the parent window to exit the touch sensitive display 110 through the second side 1354 and the child window 1350 to enter from the first side 1350 (step 1616).

[0131] The exemplary systems and methods of this disclosure have been described in relation to communication devices. However, to avoid unnecessarily obscuring the present disclosure, the preceding description omits a number of known structures and devices. This omission is not to be construed as a limitation of the scopes of the claims. Specific details are set forth to provide an understanding of the present disclosure. It should however be appreciated that the present disclosure may be practiced in a variety of ways beyond the specific detail set forth herein.

[0132] Furthermore, while the exemplary aspects, embodiments, and/or configurations illustrated herein show the various components of the system collocated, certain components of the system can be located remotely, at distant portions of a distributed network, such as a LAN and/or the Internet, or within a dedicated system. Thus, it should be appreciated, that the components of the system can be combined in to one or more devices, such as a communication device, or collocated on a particular node of a distributed network, such as an analog and/or digital telecommunications network, a packet-switch network, or a circuit-switched network. It will be appreciated from the preceding description, and for reasons of computational efficiency, that the components of the system can be arranged at any location within a distributed network of components without affecting the operation of the system. For example, the various components can be located in a switch such as a PBX and media server, gateway, in one or more

communications devices, at one or more users' premises, or some combination thereof. Similarly, one or more functional portions of the system could be distributed between a telecommunications device(s) and an associated computing device.

[0133] Furthermore, it should be appreciated that the various links connecting the elements can be wired or wireless links, or any combination thereof, or any other known or later developed element(s) that is capable of supplying and/or communicating data to and from the connected elements. These wired or wireless links can also be secure links and may be capable of communicating encrypted information. Transmission media used as links, for example, can be any suitable carrier for electrical signals, including coaxial cables, copper wire and fiber optics, and may take the form of acoustic or light waves, such as those generated during radio-wave and infra-red data communications.

[0134] Also, while the flowcharts have been discussed and illustrated in relation to a particular sequence of events, it should be appreciated that changes, additions, and omissions to this sequence can occur without materially affecting the operation of the disclosed embodiments, configuration, and aspects.

[0135] A number of variations and modifications of the disclosure can be used. It would be possible to provide for some features of the disclosure without providing others.

[0136] In other embodiments, other rules may be applied. For example, the preferred orientations of the higher and lower level related windows is different from the orientations described above.

[0137] In yet another embodiment, the systems and methods of this disclosure can be implemented in conjunction with a special purpose computer, a programmed microprocessor or microcontroller and peripheral integrated circuit element(s), an ASIC or other integrated circuit, a digital signal processor, a hard-wired electronic or logic circuit such as discrete element circuit, a programmable logic device or gate array such as PLD, PLA, FPGA, PAL, special purpose computer, any comparable means, or the like. In general, any device(s) or means capable of implementing the methodology illustrated herein can be used to implement the various aspects of this disclosure. Exemplary hardware that can be used for the disclosed embodiments, configurations and aspects includes computers, handheld devices, telephones (e.g., cellular, Internet enabled, digital, analog, hybrids, and others), and other hardware known in the art. Some of these devices include processors (e.g., a single or multiple microprocessors), memory,

nonvolatile storage, input devices, and output devices. Furthermore, alternative software implementations including, but not limited to, distributed processing or component/object distributed processing, parallel processing, or virtual machine processing can also be constructed to implement the methods described herein.

[0138] In yet another embodiment, the disclosed methods may be readily implemented in conjunction with software using object or object-oriented software development environments that provide portable source code that can be used on a variety of computer or workstation platforms. Alternatively, the disclosed system may be implemented partially or fully in hardware using standard logic circuits or VLSI design. Whether software or hardware is used to implement the systems in accordance with this disclosure is dependent on the speed and/or efficiency requirements of the system, the particular function, and the particular software or hardware systems or microprocessor or microcomputer systems being utilized.

[0139] In yet another embodiment, the disclosed methods may be partially implemented in software that can be stored on a storage medium, executed on programmed general-purpose computer with the cooperation of a controller and memory, a special purpose computer, a microprocessor, or the like. In these instances, the systems and methods of this disclosure can be implemented as program embedded on personal computer such as an applet, JAVA® or CGI script, as a resource residing on a server or computer workstation, as a routine embedded in a dedicated measurement system, system component, or the like. The system can also be implemented by physically incorporating the system and/or method into a software and/or hardware system.

[0140] Although the present disclosure describes components and functions implemented in the aspects, embodiments, and/or configurations with reference to particular standards and protocols, the aspects, embodiments, and/or configurations are not limited to such standards and protocols. Other similar standards and protocols not mentioned herein are in existence and are considered to be included in the present disclosure. Moreover, the standards and protocols mentioned herein and other similar standards and protocols not mentioned herein are periodically superseded by faster or more effective equivalents having essentially the same functions. Such replacement standards and protocols having the same functions are considered equivalents included in the present disclosure.

[0141] The present disclosure, in various aspects, embodiments, and/or configurations, includes components, methods, processes, systems and/or apparatus substantially as depicted and described herein, including various aspects, embodiments, configurations embodiments, subcombinations, and/or subsets thereof. Those of skill in the art will understand how to make and use the disclosed aspects, embodiments, and/or configurations afrer understanding the present disclosure. The present disclosure, in various aspects, embodiments, and/or configurations, includes providing devices and processes in the absence of items not depicted and/or described herein or in various aspects, embodiments, and/or configurations hereof, including in the absence of such items as may have been used in previous devices or processes, e.g., for improving performance, achieving ease and\or reducing cost of implementation.

[0142] The foregoing discussion has been presented for purposes of illustration and description. The foregoing is not intended to limit the disclosure to the form or forms disclosed herein. In the foregoing Detailed Description for example, various features of the disclosure are grouped together in one or more aspects, embodiments, and/or configurations for the purpose of streamlining the disclosure. The features of the aspects, embodiments, and/or configurations of the disclosure may be combined in alternate aspects, embodiments, and/or configurations other than those discussed above. This method of disclosure is not to be interpreted as reflecting an intention that the claims require more features than are expressly recited in each claim. Rather, as the following claims reflect, inventive aspects lie in less than all features of a single foregoing disclosed aspect, embodiment, and/or configuration. Thus, the following claims are hereby incorporated into this Detailed Description, with each claim standing on its own as a separate preferred embodiment of the disclosure.

[0143] Moreover, though the description has included description of one or more aspects, embodiments, and/or configurations and certain variations and modifications, other variations, combinations, and modifications are within the scope of the disclosure, e.g., as may be within the skill and knowledge of those in the art, afrer understanding the present disclosure. It is intended to obtain rights which include alternative aspects, embodiments, and/or configurations to the extent permitted, including alternate, interchangeable and/or equivalent structures, functions, ranges or steps to those claimed, whether or not such alternate, interchangeable and/or equivalent structures, functions, ranges or steps are disclosed herein, and without intending to publicly dedicate any patentable subject matter.

What is claimed is:

1. A method, comprising:

receiving, by an off-display gesture capture region in a communication device, a command to display a selected window;

ceasing to display, by the processor executable middleware, a currently displayed window; and

displaying, by the processor executable middleware, the selected window.

2. The method of claim 1, wherein the selected window and currently displayed window are hierarchically related and wherein a common multi-screen application outputs the currently displayed and selected windows.

3. The method of claim 2, wherein the selected window is a parent window and the currently displayed window is a child window and wherein the parent window enters a display of the communication device from a second side and the currently displayed window exits the display from an opposing first side of the communication device.

4. The method of claim 1, wherein the selected window is a child window and the currently displayed window is a parent window and wherein the parent window exits a display of the communication device from a second side and the currently displayed window enters the display from an opposing first side of the communication device.

5. The method of claim 1, wherein the selected window and currently displayed window are output by different applications and wherein the communication device comprises multiple screens, a first screen being operable and a second screen being inoperable.

6. A non-transient and tangible computer readable medium comprising processor executable instructions to perform the steps of claim 1.

7. A method, comprising:

receiving, by processor executable middleware, a command to display a first window and cease display of a second window, the first and second windows being related and being outputted by a multi-screen application; and

applying the following rules:

(i) when the first window is a parent window and the second window is a child window, the first window enters a display of the communication device from a second side and the second window exits the display from an opposing first side of the communication device;

(ii) when the first window is a child window and the second window is a parent window, the second window exits a display of the communication device from a second side and the first window enters the display from an opposing first side of the communication device.

8. The method of claim 7, wherein the command is received by an off-display gesture capture region of the communication device and wherein the communication device comprises multiple screens, a first screen being operable and a second screen being inoperable.

9. The method of claim 7, wherein rule (i) is applied.

10. The method of claim 7, wherein rule (ii) is applied.

11. A non-transient and tangible computer readable medium comprising processor executable instructions to perform the steps of claim 7.

12. A communication device, comprising:

at least first and second touch sensitive displays to display windows;

a processor; and

a computer readable memory, the computer readable memory comprising processor executable middleware operable to:

receive, by an off-display gesture capture region in a communication device, a command to display a selected window;

cease to display, by the processor executable middleware, a currently displayed window; and

display, by the processor executable middleware, the selected window.

13. The device of claim 12, wherein the selected window and currently displayed window are hierarchically related and wherein a common multi-screen application outputs the currently displayed and selected windows..

14. The device of claim 13, wherein the selected window is a parent window and the currently displayed window is a child window and wherein the parent window enters a display of the communication device from a second side and the currently displayed window exits the display from an opposing first side of the communication device.

15. The device of claim 8, wherein, when the second lower level window is opened and displayed, the first lower level window is closed automatically.

16. The device of claim 8, wherein the command is a drag or a flick, wherein the lower level window is in the predetermined touch sensitive display when the gesture is received, wherein the higher level window appears to move from the predetermined touch sensitive display to the other touch sensitive display, whereby the lower level window is displayed by the predetermined touch sensitive display and the higher level window is displayed by the other touch sensitive display.

17. The device of claim 8, wherein the multi-display communication device is in the portrait display mode and wherein the predetermined touch sensitive display is the right-most touch sensitive display.

18. The device of claim 8, wherein the selected window is a child window and the currently displayed window is a parent window and wherein the parent window exits a display of the communication device from a second side and the currently displayed window enters the display from an opposing first side of the communication device.

19. The device of claim 8, wherein the selected window and currently displayed window are output by different applications.

20. A communication device, comprising:

at least first and second touch sensitive displays to display windows;

a processor; and

a computer readable memory, the computer readable memory comprising processor executable middleware operable to:

receive a command to display a first window and cease display of a second window, the first and second windows being related and being outputted by a multi-screen application; and

apply the following rules:

(i) when the first window is a parent window and the second window is a child window, the first window enters a display of the communication device from a second side and the second window exits the display from an opposing first side of the communication device;

(ii) when the first window is a child window and the second window is a parent window, the second window exits a display of the communication device from a

second side and the first window enters the display from an opposing first side of the communication device.

21. The device of claim 20, wherein rule (i) applies.

22. The device of claim 20, wherein rule (ii) applies.

ABSTRACT

[0144] The present disclosure is directed to methodologies and devices for handling the display of hierarchically related windows in a single-screen communication device.

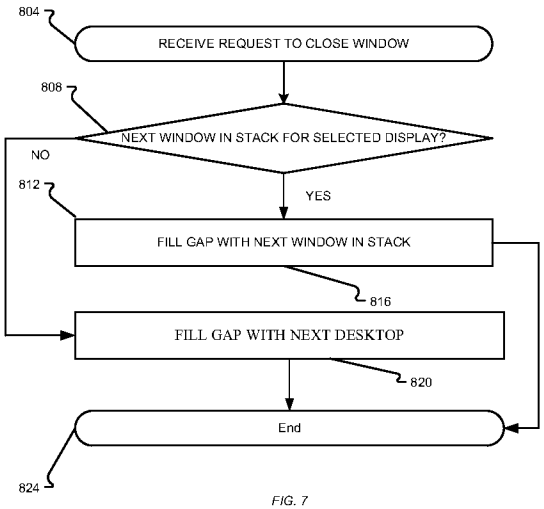


FIG. 7

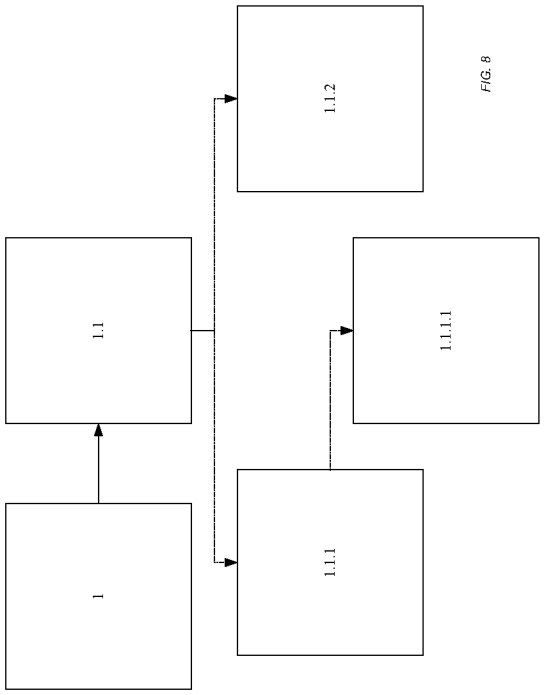


FIG. 8

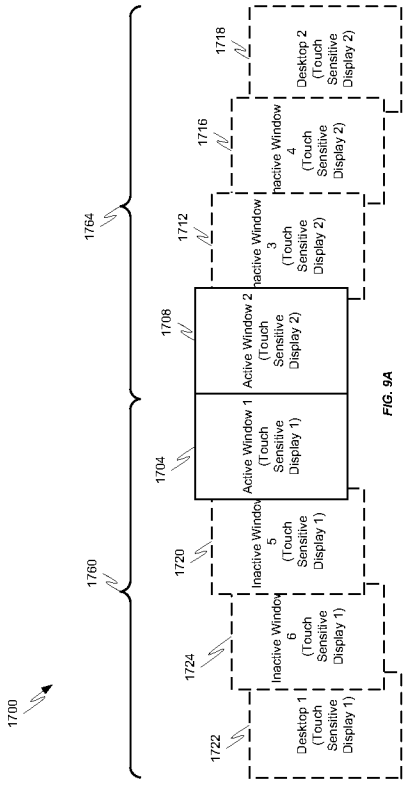


FIG. 9A

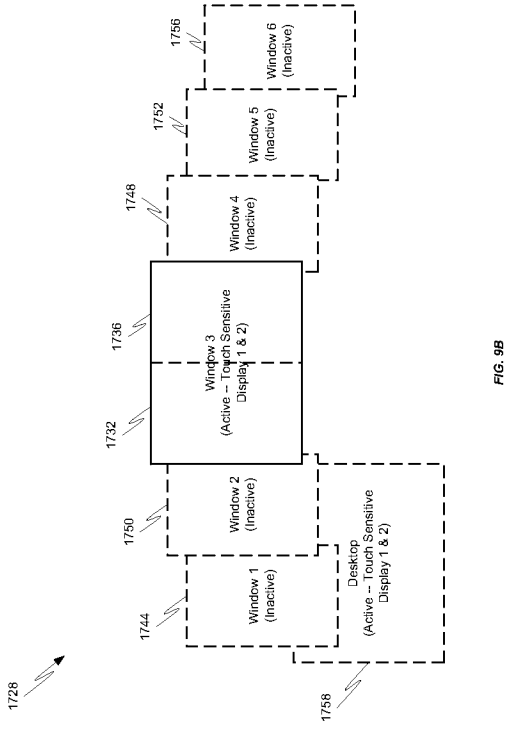


FIG. 9B

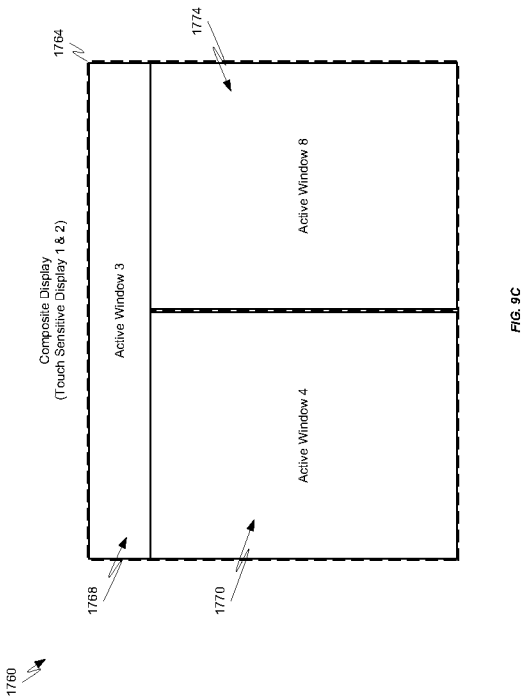


FIG. 9C

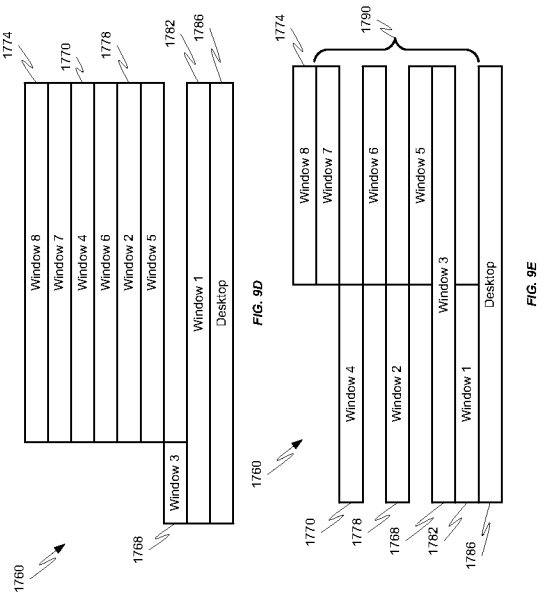


FIG. 9D

FIG. 9E

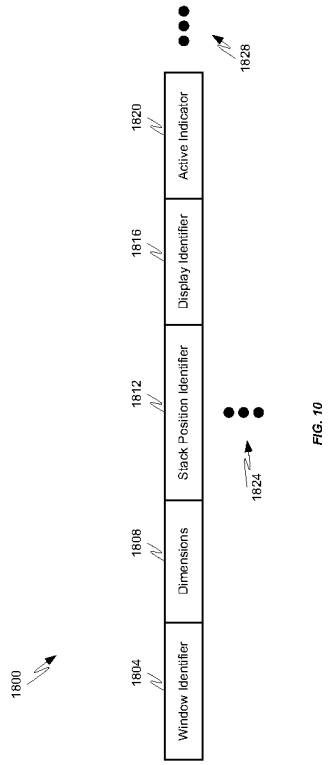


FIG. 10

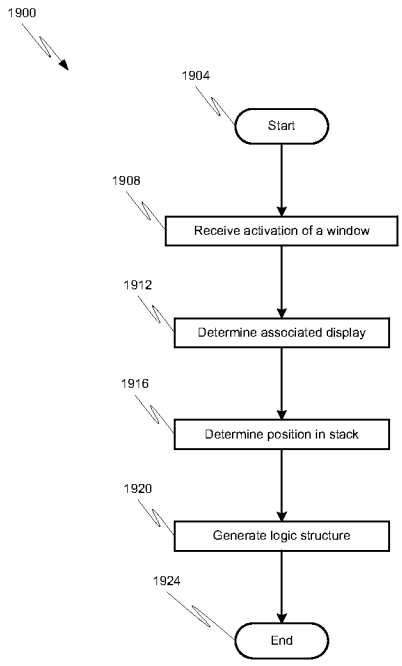


FIG. 11

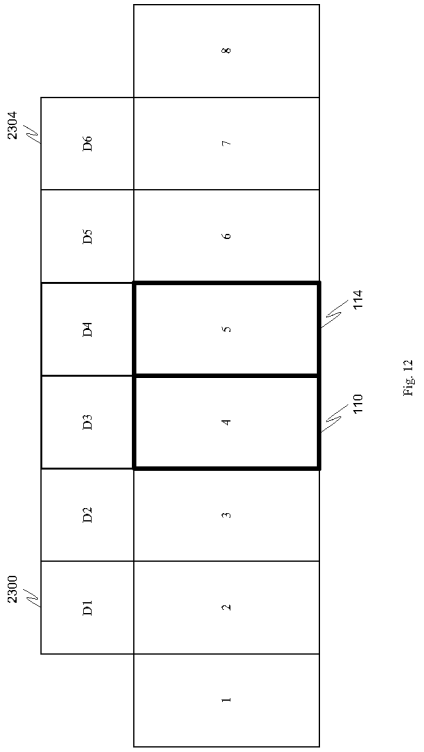


Fig. 12

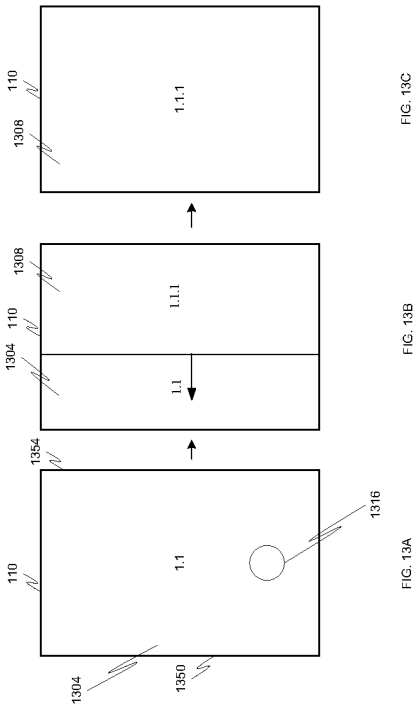


FIG. 13A FIG. 13B FIG. 13C

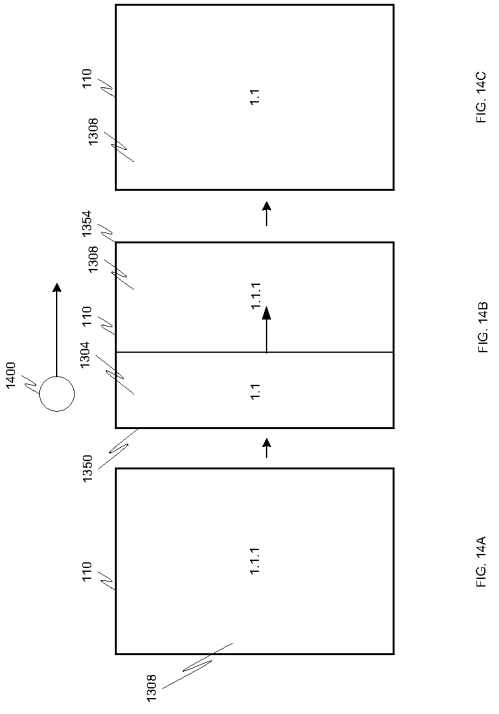


FIG. 14A FIG. 14B FIG. 14C

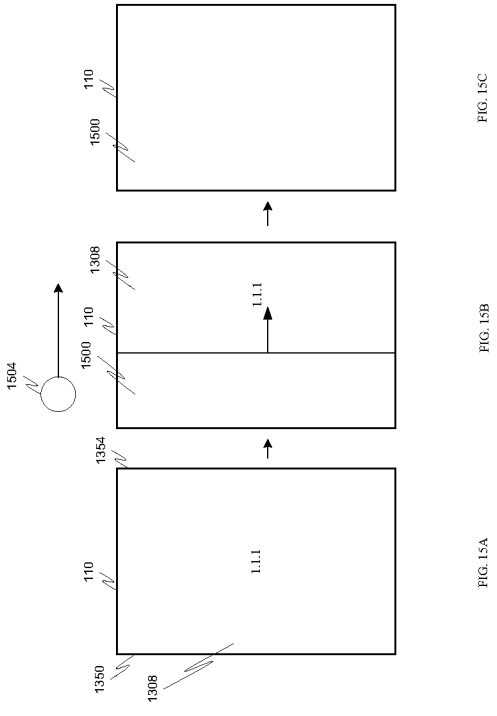


FIG. 15A FIG. 15B FIG. 15C

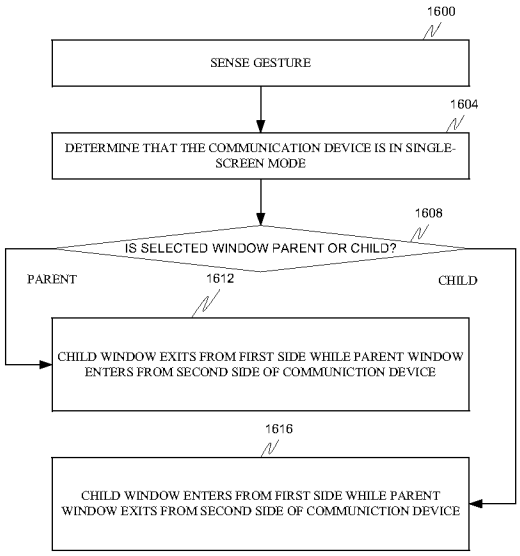

1600
SENSE GESTURE
1604
DETERMINE THAT THE COMMUNICATION DEVICE IS IN SINGLE-SCREEN MODE
1608
IS SELECTED WINDOW PARENT OR CHILD?
PARENT
1612
CHILD
CHILD WINDOW EXITS FROM FIRST SIDE WHILE PARENT WINDOW ENTERS FROM SECOND SIDE OF COMMUNICTION DEVICE
1616
CHILD WINDOW ENTERS FROM FIRST SIDE WHILE PARENT WINDOW EXITS FROM SECOND SIDE OF COMMUNICTION DEVICE

FIG. 16

METHOD AND SYSTEM FOR PERFORMING COPY-PASTE OPERATIONS ON A DEVICE VIA USER GESTURES

DESCRIPTION OF TERMS

[0001] The present section describes and/or defines terms used herein. Such terms are bolded and italicized in their respective descriptions/definitions in this "DESCRIPTION OF TERMS" section.

[0002] (1.1) The term "***application***" as used herein refers to (i) a software program installed on a device such as multi-screen device, and/or (ii) the display therefor on an electronic display screen, wherein the application is computer software designed to help the user to perform one or more predetermined tasks. Examples include enterprise software, accounting software, office suites, graphics software, and media players. An "application" has a user interface for receiving and presenting information from a user. An "application" as used herein typically refers to software (and/or its graphical presentation) wherein: (i) the application can be activated from a user input that explicitly identifies the application separately and independently from other applications; (ii) the application is interactive with the user for assisting the user in performing a task that the user is cognizant of and purposefully activates the application to perform the task, and (iii) the application is ***not*** viewed by the user as providing graphical interactions for graphically displaying other applications that are operable independently from the application for, e.g., one or more of being independently being installed, uninstalled, downloaded, displayed, not displayed, minimized or maximized, or of independently processing user input and displaying results therefrom.

[0003] (1.2) The term "***application window***" refers to a potentially complex data structure(s) defining a graphical presentation of interrelated information that can be manipulated on a graphical display screen for moving, minimizing, and maximizing as a unit. Each application window has an "application" associated therewith that defines predetermined user input thereto to which the application window is responsive according to processing performed by the associated application. As referred to herein, an "application window" also refers its graphical presentation when the application window is displayed on an electronic display device such as a "touch screen" as one skilled in the art will appreciate.

[0004] (1.3) The term "***desktop***" refers to a metaphor used to portray systems. A desktop is generally considered a "surface" that typically includes pictures, called icons, widgets, folders, etc. that can activate or show applications, windows, cabinets, files, folders, documents, and other graphical items thereon. The icons are generally selectable to initiate a task through user interface interaction to allow a user to execute applications or conduct other operations. More precisely, as used herein a "desktop" refers to a potentially complex data structure for providing graphical interactions for graphically displaying with a display of the desktop a plurality of applications such that by user input to the desktop, the displays of the applications can be rearranged relative to the desktop display, added to the desktop display, deleted from the desktop display, and activated from the desktop display. Each desktop includes or has associated therewith data identifying it as a desktop and not an application and not an application window. Similarly, each application and application window includes or has associated therewith data identifying it has an application or application window and not a desktop.

SUMMARY

[0005] There is a need for a multi-display handheld computing device that provides for enhanced power and/or versatility compared to conventional single display handheld computing devices. In particular, using the terms described hereinabove, the present disclosure is directed to copying data from one application display location, or screen, to a display location of another such application; e.g., a copy-paste operation as this term is used in the art. More particularly, the present disclosure is directed to performing such a copy-paste operation using gestures detected on the surface of one or more touch sensitive screen displays.

[0006] In one embodiment, the method of the copy-paste operation disclosed herein includes detecting that a user is using his/her finger(s) (or other means, e.g., a stylus) to perform a gesture such as a "press and hold" gesture, wherein the user presses and holds (e.g., for at least two seconds) at a desired location on a touch sensitive screen display for identifying underneath the "press and hold" location, the information that the user desires to

copy. Upon detecting this "press and hold" user gesture and the display location therefor, the device determines a displayed informational content associated with the location (more particularly, associated with an application window display area/frame referred to herein as a "source area") that is appropriate for copying. A clipboard utility may be provided by the device for copying the selected information from the source area. Subsequently, upon the user inputting additional user gestures to the touch sensitive display screen(s), e.g., by dragging his/her finger gesture configuration thereon, the user can designate to the device an application window (more particularly, an application window display area/frame therein, referred to herein as a "target area") in which the selected information is pasted therein.

[0007] For a multi-display handheld computing device, the copy-paste operation may be performed over multiple touch sensitive display screens, wherein the information from an identified source area is on a first screen display, and the user identified target area is on a second screen display. Moreover, in performing such an inter-screen copy-paste operation (via gestures), the configuration of the display screens' relative physical orientation to one another may be taken into account. In particular, since the multi-display device may allow the display screens to be folded relative to one another, or alternatively, unfolded such that each of the display screens face a same direction, such inter-screen copy-paste may occur only when the the display screens are unfolded relative to one another.

[0008] In a further embodiment, the source area (and/or data therein) may be highlighted during its selection process in order to provide the user with appropriate visual feedback that the desired informational content has been selected for the copy-paste operation.

[0009]

[0010] These and other needs are addressed by the various aspects, embodiments, and/or configurations of the present disclosure. Also, while the disclosure is presented in terms of exemplary embodiments, it should be appreciated that individual aspects of the disclosure can be separately claimed.

[0011] In one embodiment, the present disclosure describes a method for performing a copy-paste operation using user finger gesture input to a device having at least one display screen, comprising the steps (a) through (c) following:

[0012] (a) receiving a first finger gesture input to one of the touch sensitive display screens;

[0013] (b) selecting a source area of the touch sensitive display screens, the source area selected identified by a display screen location for the first finger gesture input;

[0014] (c) receiving a finger drag gesture input for identifying a target area of the touch sensitive display screens into which data from the source data is to be copied, wherein the target area corresponds to a location on the touch sensitive display screens where the drag gesture is last detected before it ceases to be detected; and

[0015] (d) copying the data to the target area.

[0016] In a related embodiment, the present disclosure describes the multi-display device as having the following elements (a) through (d):

[0017] (a) first and second display screens of the device, wherein the device includes a folding mechanism attached to each of the first and second display screens for providing the first and seconds display screens in a folded configuration wherein the first and second display screens face in opposite directions, and in an unfolded configuration wherein the first and second display screens face in a substantially same direction;

[0018] (b) a multi-display manager for determining that the first and second display screens are in an unfolded configuration;

[0019] (c) a gesture interpreter for interpreting each of a first finger gesture input to the first display screen, and a finger drag gesture input to each of the first and second display screens;

[0020] (d) a window manager for receiving: (i) first data from the gesture interpreter, the first data indicative of the first finger gesture input, wherein the first data is for identifying a source area, displayed on the first display screen, from which displayed data is to be copied, and (ii) second input from the gesture interpreter, the second data indicative of the finger drag gesture input, wherein the second data is for identifying a target area of the second display screen into which data from the source data is to be copied, wherein the target area corresponds to a location of the second display screen where the drag gesture is last detected by the gesture interpreter before the drag gesture ceases to be detected by the gesture interpreter;

[0021] wherein when the source area and the target area are respectively determined on the first and second display screens, data representative of a contents of the source area is copied into the target area.

[0022]

[0023] The present disclosure can provide a number of advantages depending on the particular aspect, embodiment, and/or configuration. In particular, the above described copy-paste operation as provided via user gesture input on a multi-display device provide an more intuitive and enhanced user experience for a user interacting with an embodiment of the multi-display device. These and other advantages will be apparentfrom the disclosure.

Description of Drawings

[0024] FIG. 7A representation of a logical window stack;

[0025] FIG. 7B is another logical window stack;

[0026] FIG. 7C is another representation of an embodiment of a logical window stack;

[0027] FIG. 7D is another representation of an embodiment of a logical window stack;

[0028] FIG. 7E is another representation of an embodiment of a logical window stack;

[0029] FIG. 8 is block diagram of an embodiment of a logical data structure for a window stack;

[0030] FIG. 9 is a flow chart of an embodiment of a method for creating a window stack;

[0031] FIGS. 10A through 10D show an illustration of a copy-paste operation performed via finger gestures, wherein the copy-paste operation copies from an application 564 displayed on display 114, and the paste task is into the display of another application 564 displayed on the display 110;

[0032] FIG. 11 is a flowchart providing high level steps for performed an embodiment of the copy-operation.

[0100] The device 100 manages desktops and/or windows with at least one window stack 700, 728, as shown in Figs. 7A and 7B. A window stack 700, 728 is a logical arrangement of active and/or inactive windows for a multi-screen device. For example, the window stack 700 or728 may be logically similar to a deck of cards, where one or more windows or desktops are arranged in order, as shown in Figs. 7A and 7B. An active window is a window that is currently being

displayed on at least one of the touch sensitive displays 110, 114. For example, windows displayed on the touch sensitive displays 104 and 108 are active windows. An inactive window is a window that was opened and displayed but is now "behind" an active window and not being displayed. In embodiments, an inactive window may be for an application 564 that is suspended, and thus, the window is not displaying active content. For example, windows 712, 716, 720, and 724 (Fig. 7A) are inactive windows.

[0101] A window stack 700, 728 may have various arrangements or organizational structures. In the embodiment shown in Fig. 7A, the device 100 includes a first stack 760 associated with the first touch sensitive display 110 and a second stack 764 associated with the second touch sensitive display 114. Thus, each touch sensitive display 110 and 114 can have an associated respective window stack 760, 764. These two window stacks 760, 764 may have different numbers of windows arranged in the respective stacks 760, 764. Further, the two window stacks 760, 764 can also be identified differently and managed separately. Thus, the first window stack 760 can be arranged in order from a first window 704 to a next window 720 to a last window 724 and finally to a desktop 722, which, in embodiments, is at the "bottom" of the window stack 760. In embodiments, the desktop 722 is not always at the "bottom" since application windows can be arranged in the window stack below the desktop 722, and the desktop 722 can be brought to the "top" of a stack over other windows as a result of performing a desktop reveal operation. Likewise, the second stack 764 can be arranged from a first window 708 to a next window 712 to a last window 716, and finally to a desktop 718, which, in embodiments, is a single desktop area, with desktop 722, under all the windows in both window stack 760 and window stack 764. A logical data structure for managing the two window stacks 760, 764 may be as described in conjunction with Fig. 8 hereinbelow.

[0102] Another arrangement for a window stack 728 is shown in Fig. 7B. In this embodiment, there is a single window stack 728 for both touch sensitive displays 110, 114. Thus, the window stack 728 is arranged from a desktop 758 to a first window 744 to a last window 756. A window can be arranged in a position among all windows without an association to a specific touch sensitive display 110, 114. In this embodiment, each window is in the order of windows. Further, at least one window is identified as being active. For example, a single window may be rendered in two portions 732 and 736 that are displayed, respectively, on the first touch sensitive

screen 110 and the second touch sensitive screen 114. This single window may only occupy a single position in the window stack 728 although it is displayed on both displays 110, 114.

[0103] Yet another arrangement of a window stack 762 is shown in Figs. 7C through 7E. The window stack 762 is shown in three "elevation" views. In Fig. 7C, the top of the window stack 762 is shown. Two sides of the window stack 762 are shown in Figs. 7D and 7E. In this embodiment, the window stack 762 resembles a stack of bricks. The windows are stacked on each other. Looking from the top of the window stack 762 in Fig, 7C, only the top most windows in the window stack 762 are seen in different portions of the composite display 766. The composite display 766 represents a logical model for the entire display area of the device 100, which can include touch sensitive display 110 and touch sensitive display 114. A desktop 786 (Figs. 7D and 7E) or a window can occupy part or all of the composite display 766.

[0104] In the embodiment shown, the desktop 786 is the lowest display or "brick" in the window stack 762. Thereupon, window 1 782, window 2 782, window 3 768, and window 4 770 are layered. Window 1 782, window 3 768, window 2 782, and window 4 770 only occupy a portion of the composite display 766. Thus, another part of the stack 762 includes window 8 774 and windows 5 through 7 shown in section 790. Only the top window in any portion of the composite display 766 is actually rendered and displayed. Thus, as shown in the top view in Fig. 7C, window 4 770, window 8 774, and window 3 768 are displayed as being at the top of the display in different portions of the window stack 762. A window can be dimensioned to occupy only a portion of the composite display 766 to "reveal" windows lower in the window stack 760. For example, window 3 768 is lower in the stack than both window 4 770 and window 8 774 but is still displayed. A logical data structure to manage the window stack can be as described in conjunction with Fig. 8.

[0105] When a new window is opened on the device 100, the newly activated window is generally positioned at the top of the stack. However, where and how the window is positioned within the stack can be a function of the orientation of the device 100, the context of what programs, functions, software, etc. are being executed on the device 100, how the stack is positioned when the new window is opened, etc. To insert the window in the stack, the position in the stack for the window is determined and the touch sensitive display 110, 114 to which the window is associated may also be determined. With this information, a logical data structure for the window can be created and stored. When user interface or other events or tasks change the

arrangement of windows, the window stack(s) can be changed to reflect the change in arrangement. It should be noted that these same concepts described above can be used to manage the one or more desktops for the device 100.

[0106] A logical data structure 800 for managing the arrangement of windows or desktops in a window stack is shown in Fig. 8. The logical data structure 800 can be any data structure used to store data whether an object, record, file, etc. The logical data structure 800 can be stored in any type of database or data storage system, regardless of protocol or standard. In embodiments, the logical data structure 800 includes one or more portions, fields, attributes, etc. that store data in a logical arrangement that allows for easy storage and retrieval of the information. Hereinafrer, these one or more portions, fields, attributes, etc. shall be described simply as fields. The fields can store data for a window identifier 804, dimensions 808, a stack position identifier 812, a display identifier 816, and/or an active indicator 820. Each window in a window stack can have an associated logical data structure 800. While only a single logical data structure 800 is shown in Fig. 8, there may be more or fewer logical data structures 800 used with a window stack (based on the number of windows or desktops in the stack), as represented by ellipses 824. Further, there may be more or fewer fields than those shown in Fig. 8, as represented by ellipses 828.

[0107] A window identifier 804 can include any identifier (ID) that uniquely identifies the associated window in relation to other windows in the window stack. The window identifier 804 can be a globally unique identifier (GUID), a numeric ID, an alphanumeric ID, or other type of identifier. In embodiments, the window identifier 804 can be one, two, or any number of digits based on the number of windows that can be opened. In alternative embodiments, the size of the window identifier 804 may change based on the number of windows opened. While the window is open, the window identifier 804 may be static and remain unchanged.

[0108] Dimensions 808 can include dimensions for a window in the composite display 766. For example, the dimensions 808 can include coordinates for two or more corners of the window or may include one coordinate and dimensions for the width and height of the window. These dimensions 808 can delineate what portion of the composite display 766 the window may occupy, which may be the entire composite display 766 or only part of composite display 766. For example, window 4 770 may have dimensions 880 that indicate that the window 4 770 will

occupy only part of the display area for composite display 766, as shown in Fig. 7C. As windows are moved or inserted in the window stack, the dimensions 808 may change.

[0109] A stack position identifier 812 can be any identifier that can identify the position in the stack for the window or may be inferred from the window's control record within a data structure, such as a list or a stack. The stack position identifier 812 can be a GUID, a numeric ID, an alphanumeric ID, or other type of identifier. Each window or desktop can include a stack position identifier 812. For example, as shown in the embodiment of Fig. 7A, window 1 704 in stack 1 760 can have a stack position identifier 812 of 1 identifying that window 704 is the first window in the stack 760 and the active window. Similarly, window 6 724 can have a stack position identifier 812 of 3 representing that window 724 is the third window in the stack 760. Window 2 708 can also have a stack position identifier 812 of 1 representing that window 708 is the first window in the second stack 764. However, in Fig. 7B, an alternative stack position numbering of windows is provided; i.e., window 1 744 can have a stack position identifier 812 of 1, window 3, rendered in portions 732 and 736, can have a stack position identifier 812 of 3, and window 6 756 can have a stack position identifier 812 of 6. Thus, depending on the type of stack, the stack position identifier 812 can represent a window's location in the stack.

[0110] A display identifier 816 can identify that the window or desktop is associated with a particular display, such as the first display 110 or the second display 114, or the composite display 760 composed of both displays. While this display identifier 816 may not be needed for a multi-stack system, as shown in Fig. 7A, the display identifier 816 can indicate whether a window in the serial stack of Fig. 7B is displayed on a particular display. Thus, window 3 may have two portions 732 and 736 in Fig. 7B. The first portion 732 may have a display identifier 816 for the first display while the second portion 736 may have a display identifier 816 for the second display 114. However, in alternative embodiments, the window may have two display identifiers 816 that represent that the window is displayed on both of the displays 110, 114, or a display identifier 816 identifying the composite display. In another alternate embodiment, the window may have a single display identifier 816 to represent that the window is displayed on both of the displays 110, 114.

[0111] Similar to the display identifier 816, an active indicator 820 may not be needed with the dual stack system of Fig. 7A, as the window in stack position 1 is active and displayed. In the system of Fig. 7B, the active indicator 820 can indicate which window(s) in the stack is being

displayed. Thus, window 3 may have two portions 732 and 736 in Fig. 7B. The first portion 732 may have an active indicator 820 while the second portion 736 may also have an active indicator 820. However, in alternative embodiments, window 3 may have a single active indicator 820. The active indicator 820 can be a simple flag or bit that represents that the window is active or displayed.

[0112] An embodiment of a method 900 for creating a window stack is shown in Fig. 9. While a general order for the steps of the method 900 is shown in Fig. 9. Generally, the method 900 starts with a start operation 904 and ends with an end operation 928. The method 900 can include more or fewer steps or can arrange the order of the steps differently than those shown in Fig. 9. The method 900 can be executed as a set of computer-executable instructions executed by a computer system and encoded or stored on a computer readable medium. Hereinafter, the method 900 shall be explained with reference to the systems, components, modules, software, data structures, user interfaces, etc. described in conjunction with Figs. 1-8.

[0113] A multi-screen device 100 can receive activation of a window, in step 908. In embodiments, the multi-screen device 100 can receive activation of a window by receiving an input from the touch sensitive display 110 or 114, the configurable area 112 or 116, a gesture capture region 120 or 124, or some other hardware sensor operable to receive user interface inputs. The processor may execute the Task Management Module 540 (Fig. 5A) to receive the input. The Task Management Module 540 can interpret the input as requesting an application task to be executed that will open a window in the window stack.

[0114] In embodiments, the Task Management Module 540 places the user interface interaction in the task stack 552 (Fig. 5A) to be acted upon by the Display Configuration Module 568 (Fig. 5B) of the Multi-Display Management Module 524 (Figs. 5A, 5B). Further, the Task Management Module 540 waits for information from the Multi-Display Management Module 524 to send instructions to the Window Management Module 532 to create the window in the window stack.

[0115] The Multi-Display Management Module 524, upon receiving instruction from the Task Management Module 540, determines to which touch portion of the composite display 760, the newly activated window should be associated, in step 912. For example, window 4 770 is associated with the composite display 766 (Fig. 7A). In embodiments, the device state module 574 (Fig. 5B) of the Multi-Display Management Module 524 may determine how the device is

oriented or in what state the device is in, e.g., open, closed, portrait, etc. Further, the preferences module 572 (Fig. 5B) and/or requirements module 580 (Fig. 5B) may determine how the window is to be displayed. The gesture module 576 (Fig. 5B) may determine the user's intentions about how the window is to be opened based on the type of gesture and the location of where the gesture is made.

[0116] The Display Configuration Module 568 (Fig. 5B) may use the input from these modules and evaluate the current window stack 760 to determine the best place and the best dimensions, based on a visibility algorithm, to open the window. Thus, the Display Configuration Module 568 determines the best place to put the window at the top of the window stack 760, in step 916. The visibility algorithm, in embodiments, determines for all portions of the composite display, which windows are at the top of the stack. For example, the visibility algorithm determines that window 3 768, window 4 770, and window 8 774 are at the top of the stack 760 as viewed in Figs. 7C through 7E. Upon determining where to open the window, the Display Configuration Module 568 can assign a display identifier 816 and possibly dimensions 808 to the window. The display identifier 816 and dimensions 808 can then be sent back to the Task Management Module 540. The Task Management Module 540 may then assign the window a stack position identifier 812 indicating the windows position at the top of the window stack.

[0117] In embodiments, the Task Management Module 540 sends the window stack information and instructions to render the window to the Window Management Module 532 (Fig. 5A). The Window Management Module 532 and the Task Management Module 540 (Fig. 5A) can create the logical data structure 800, in step 924. Both the Task Management Module 540 and the Window Management Module 532 may create and manage copies of the window stack. These copies of the window stack can be synchronized or kept similar through communications between the Window Management Module 532 and the Task Management Module 540. Thus, the Window Management Module 532 and the Task Management Module 540, based on the information determined by the Multi-Display Management Module 524, can assign dimensions 808, a stack position identifier 812 (e.g., window 1 782, window 4 770, etc.), a display identifier 816 (e.g., touch sensitive display 1 110, touch sensitive display 2 114, composite display identifier, etc,), and an active indicator 820, which is generally always set when the window is at the "top" of the stack. The logical data structure 800 may then be stored

by both the Window Management Module 532 and the Task Management Module 540. Further, the Window Management Module 532 and the Task Management Module 540 may thereinafter manage the window stack and the logical data structure(s) 800.

[0118] In at least one embodiment of the device 100, the MDM 524 (and its Gesture Module 576) may be operable for assisting the Window Management Module 532 in copying and pasting the contexts of one application 564 to another via user gestures. In particular, to copy data from one location, or screen, to another the user presses and holds an area on the screen where the data that is desired to be copied is located; the data is then automatically highlighted and selected, then the user can drag the highlighted data to a destination location via finger gestures. This drag operation can cross over a first screen to a second screen where the data may be pasted. Figs. 10A through 10D is illustrative of the present copy and paste feature of an embodiment of the device 100. Referring to Fig. 10A, the device 100 is shown in an open portrait state, wherein the display 110 displays a email application window 1004 for composing a new email to "Adam". Note that the content area 1008 of the email currently states "Here is the text:". The display 114 shows an illustrative presentation of an Internet browser window 1012, wherein the URL for the webpage being displayed is "www.flextronics.com", and at least a portion of the contents of the webpage is shown in the browser display area/frame 1016 In Fig. 10B, a user inputs a user gesture 1020 into the display area/frame 1016 for selecting the contents thereof. In the presently illustrated embodiment, the user gesture 1020 is a two finger press and hold gesture (e.g., a two finger "long press" 424, Fig. 4B), wherein the user presses with two fingers and maintains the finger pressure on the display with both fingers for, e.g., at least two seconds for thereby selecting the contents of the display area/frame 1016 In Fig. 10B, this two finger press and hold gesture 1020 is represented by the pair of concentric circles 1024. However, it is within the scope of the present disclosure that other/alternative user gestures may be used to select the contents of a display area/frame, e.g., 1016. For example, instead (or in addition to) the Gesture Module 576 being programmed to recognize the two finger press and hold gesture 1020 for selecting the contents of a display area/frame, the Gesture Module may be programmed to recognize a single finger press and hold combined with a finger tap 420 for selecting such contents. Note that the user could have selected the URL area/frame 1028 instead of the area/frame 1016 if the user gesture 1020 would have been applied to the URL area/frame. Further note that in order for the Window

Management Module 532 to identify the particular area/frame in which the user gesture 1020 is being applied, the MDM, it may be necessary for the MDM to not only interrogate the window stack for the display (110 or 114) to which the user gesture is being input (for determining, e.g., the window in which the gesture is being input), but also interrogate the display data structure for the application window selected for determining the (any) frame within the application window to select.

[0119] Once the Gesture Module 576 recognizes the user gesture 1020 for selecting the area/frame of an application window (e.g., window 1012), the selected area/frame may be highlighted (or alternatively, only its border may be highlighted) on its display (110 or 114) so that the user receives feedback as to what portion of the screen display has been (or will be) copied into another application window. In Fig. 10B, border of the browser display area/frame 1016 is identified by a thick line width to indicate that this display area/frame has been selected by the Window Management Module 532.

[0120] Note that for hierarchical display areas/frames, the first area/frame selected by the Window Management Module 532 may be larger or smaller than what the user desires to copy. Accordingly, embodiments may allow the user to input additional gestures for selecting the desired portion of a display area/frame. For example, the user may cease the press and hold gesture with one of his/her fingers, and use this finger to quickly tap the display (110 or 114) for selecting another display area/frame within the hierarchy of display areas/frames, which would be highlighted instead of the previously selected display area/frame from the hierarchy. Thus, the user may precisely select the content to be copied into the display of another application 564.

[0121] Once the desired display area/frame is selected for copying, the user may drag his/her fingers that are pressing on the display area/frame (e.g., 1016) to be copied so that the termination of the drag identifies the window (or display area/frame thereof) of the target application 564 in which the selected information is to be copied. Fig. 10C is illustrative of this. In particular, drag gesture 1030 is a representation of a drag of the user gesture 1020, wherein Fig. 10C further illustrates the user dragging of the user gesture 1020 from the display 114 to the drag termination location 1032 in the display 110 thereby designating to the Window Management Module 532 the target display area/frame 1008 in which to copy the information selected from display area/frame 1016. Note that the drag gesture 1030 may commence by moving the gesture 1020 across the display 100 substantially without releasing contact for

touching the display 110; e.g., it may be determined that there is substantially no release of display 110 contact in transitioning from the gesture 1020 to the drag gesture 1030 if it is detected at least every 0.1 milliseconds. Moreover, in one embodiment, the selected information is copied as it resides in memory 508; e.g., text is copied as text data, pictures are copied as picture data so that data manipulations can be performed (to the degree such are supported) in the target application 564. Thus, the device 100 allows displayed information (e.g., image, text, etc.) to be selected with a simple user gesture and dragged to a new location without requiring additional user steps or lifting of the user's fingers or stylus from the displays 110 and 114. Moreover, the copied information can be copied and pasted to a different display 110 or 114 of the device by the drag user gesture 1030 extending onto both the displays 110 and 114. Note that in one embodiment, the drag user gesture may be accompanied with a corresponding drag of an image of the information to be pasted into the target application 564. In particular, the drag image may be generated as a reduced size replica of the selected information, and/or as a semi-transparent replica of the selected information wherein the user performing the drag can both recognize what is being dragged (for pasting), and identify the displayed presentations over which the drag image is being dragged.

[0122] Accordingly, Fig. 10D shows the resulting configuration of the displays 110 and 114 afrer the text of browser display area/frame 1016 is copied and pasted into the email application window 1004 (and more particularly, into the email content area 1008. Note that browser display area/frame 1016 is not longer highlighted since the copy-paste operation has been completed.

[0123] The flowchart of Fig. 11 shows illustrative high level steps that may be performed by an embodiment of the device 100 in performing a copy-paste operation as described hereinabove with reference to Figs. 10A-10D. However, before describing the flowchart steps, it noteworthy that in order for the copy-paste operation to be eligible for being performed wherein the user drag gesture 1020 extends from one of the displays 110, 114 to the other of these displays, the device 100 must in either the open portrait or open landscape state as determined by the MDM 524. Otherwise, the copy-paste operation is confined to a single one of displays 110 or 114. Accordingly, in step 1104, the Gesture Module 576 determines that a user input to one of the displays 110 or 114 is a user gesture (e.g., user gesture 1020) for at least selecting a display area/frame of an application window. In step 1108, upon receiving input from the Gesture

Module 576 indicating that such a user gesture has been detected as occurring on one of the displays 110 or 114, the Window Management Module 532 determines a display area/frame ("source area" herein) of the window application where the user gesture was first detected, wherein the source area is determined by the user as the desired area/frame to copy. In performing the step 1108, the Window Management Module 532 may, in one embodiment, receive additional inputs from the Gesture Module 576 for notifying the Window Management Module 532of additional user gesture input for identifying various display areas/frames in a display hierarchy of such display areas/frames. For example, the user may input the user gesture 1020 upon a displayed picture on a webpage, and the webpage may be initially identified by the Window Management Module 532. Thus, the Window Management Module 532 may cause the webpage (or a frame boundary thereof) to be highlighted to the user thereby indicating that the webpage has been selected for coping. Accordingly, the user may provide an additional user gesture (e.g. a finger tap 400 as described above with reference to Fig. 10B) to indicate that another area/frame of the display is desired to be copied. Accordingly, assuming there is no intermediate display area/frame in a hierarchy that ends in the picture, upon the Window Management Module 532 receiving data from the Gesture Module 576 indicating that the additional user gesture has been input by the user, the MDM selects the picture, and highlights this picture on the display (110 or 114) to confirm to the user the information selected to be copied.

[0124] Thus, as indicated in the example of selecting a displayed picture above, the Window Management Module 532 may perform the step 1112 for highlighting the display area/frame that has been selected for copying. Following this step, upon receiving additional input from the Gesture Module 576 regarding (any) movement (or other change) in the user gesture 1020, in step 1116, the Window Management Module 532 receives and stores a display location for each such additional input. Note that previous display locations corresponding to a previous location of the user gesture 1020 may be overwritten with each new display location. Moreover, the Window Management Module 532 may be programmed to create a semi-transparent copy (or, e.g., a non-transparent iconic representation) of the selected information which may be displayed as following beneath the dragging of the user gesture 1020 about the displays 110 and/or 114. Furthermore, for each input the Window Management Module 532 receives of the user gesture 1020 display location, the MDM may in real time: (i) determine a (any) display area/frame

identified by the user gesture display location, and once determined, (ii) highlight or otherwise identify on the display(s) 110 and/or 114 such a determined display area/frame (or the border thereof) as being a candidate display area/frame that can accept the selected information in a paste operation. Note that in determining (i) immediately above, if the user gesture display location identifies a hierarchical collection of display areas/frames, then the Window Management Module 532 may determine the lowest ordered (e.g., smallest) display area/frame in the hierarchy that can accept the selected information. Thus, for example, if the content area 1008 of Figs. 10 included a form with entries that can accept user input, then if a first entry was a check box, then if the user gesture 1020 were dragged over this first entry, it would not be identified as appropriate for receiving the selected information, and instead, a higher ordered display area/frame would be determined, e.g., the content area 1008. If, however, a second form entry is determined to be beneath the current location of the user gesture 1020, and this second form entry is able to accept the selected information for the copy-paste operation (e.g., textual webpage input as selected for the copy-paste operation in Fig. 10B), then the area/frame for this second form entry would be identified (e.g., via highlighting) on its display(s) 110 or 114 as a candidate for receiving a paste of the selected information.

[0125] Assuming the MDM 524 has informed the Gesture Module 576 that the device 100 in an open (unfolded) configuration, then when the user gesture 1020 drags off its display 110, 114, the Gesture Module 576 may attempt to correlate a corresponding drag gesture 1020 on the other one of the displays 110, 114. In particular, assuming the device 100 is unfolded (the Gesture Module 576 has been informed of such), the Gesture Module 576 may detect that the gesture 1020 has lefr its display on a side between the displays 110, 114 (rather than some other side which may be interpreted as a termination of the copy-paste operation). Accordingly, the Gesture Module 576 may wait a predetermined time of, e.g., at most 0.5 seconds for detecting a corresponding drag gesture 1020 near the border of the other of the displays 110, 114, wherein this border is closest to the border of the display 110, 114 from which the drag gesture 1020 ceased to be detected. Thus, the device 100 can track the user drag gesture 1020 from one of the displays 110, 114 to the other of the displays, wherein the drag gesture 1020 substantially continuously traverses from one of the displays 110, 114 to the other display.

[0126] In step 1120, upon the Window Management Module 532 receiving input from the Gesture Module 576 that the user gesture 1020 has ceased to be detected on one of the displays

110 or 114, the MDM determines whether there is an area/frame underneath the last stored display location of the user gesture 1020, wherein this area/frame (the "target area" herein) accepts input as provided by the source area. If not, then there is no appropriate area/frame in which to paste the selected information, and accordingly the process corresponding to flowchart of Fig. 11 terminates at 1124 without performing the copy-paste operation. Note that a user can use this technique to safely terminate the copy-paste operation without causing a paste of the selected information. That is, the user could terminate the drag of the user gesture 1030 over a display location that does not identify an area/frame of the displays 110 or 114 that accepts the selected information of the source area as input. However, in an additional/alternative embodiment, a user may terminate the copy-paste operation without completing the paste by accelerating the user gesture 1030 off the displays 110 and 114.

[0127] If, on-the-other-hand, step 1120 results in identifying a display area/frame of an application window that can accept the source area input, then in step 1128, the selected contents of source area is copied to a temporary data storage (not separately shown) of memory 508 (e.g., a "clipboard" as one skilled in the art will understand.).

[0128] Subsequently, in step 1132, this source area information that is newly entered in the temporary data storage is then automatically (e.g., without requiring user input) copied into the target area by the Window Manager Module 532. Note that for applications 564 in which, e.g., text, tables, pictures, videos, and/or combinations thereof may be input, such applications are likely to be configured when installed in the device 100 so that these applications can accept input from a clipboard as part of the device's application support services provided by the application manager 562. Moreover, note that the paste of the source area information into the target may require data type conversions so that an appropriate version of the selected content of the source area can be pasted into the target area. For example, if the selected content of the source area is Microsoft Word® text, and the target area the Google Chrome® then the selected content may need to be converted to a "rich text" format as one skilled in the art will understand.

[0129] Subsequently, in step 1140, if there are highlighted portions of the displays 110,114 due to the present copy-paste operation, such highlighting is removed or turned off by the Window Management Module 532. In particular, the Window Management Module 532 may turn off any highlighting of the source and target areas. Subsequently, the flowchart of Fig. 11 terminates at 1144.

What is claimed is:

1. A method for performing a copy-paste operation using user finger gesture input to a device having one or more touch sensitive display screens, comprising:

receiving a first finger gesture input to one of the touch sensitive display screens;

selecting a source area of the touch sensitive display screens, the source area selected identified by a display screen location for the first finger gesture input;

receiving a finger drag gesture input for identifying a target area of the touch sensitive display screens into which data from the source data is to be copied, wherein the target area corresponds to a location on the touch sensitive display screens where the drag gesture is last detected before it ceases to be detected; and

copying the data to the target area.

2. The method of Claim 1, wherein the one touch sensitive display screen includes a capacitive touch sensitive area for receiving the finger gesture input.

3. The method of Claim 1, further including a step of outputting on one of the touch sensitive display screens a change in the display of the source area for identifying the source area to a user.

4. The method of Claim 3, further including a step of selecting the target area by: (i) determining the target area by identifying the location on one of the touch sensitive display screens where the drag gesture is last detected before it ceases to be detected, and (ii) determining that the target area is configured to receive the data.

5. The method of Claim 4, wherein the step of selecting includes determining that the target area that accepts a clipboard input.

6. The method of Claim 1, wherein the first finger gesture includes pressing and holding a finger against the one touch sensitive display screen for a predetermined time for identifying a display location for the source area.

7. The method of Claim 6, wherein the finger drag gesture includes dragging the finger gesture on the one touch sensitive display screen.

8. The method of Claim 1, further including identifying to a user one or more display areas, of the touch sensitive display screens, as the user inputs a portion of the drag gesture over each of the one or more display areas that are determined to be acceptable for pasting the data therein.

9. The method of Claim 1, wherein the finger drag gesture extends across a boundary of a first of the touch sensitive display screens and onto a second of the touch sensitive display screens, wherein the first and second display screens are foldable relative to one another.

10. A computer readable medium for performing a copy-paste operation using user finger gesture inputs to first and second display screens of a device, wherein the device includes a folding mechanism attached to each of the first and second display screens for providing the first and seconds display screens in a folded configuration wherein the first and second display screens face in opposite directions, and in an unfolded configuration wherein the first and second display screens face in a substantially same direction, comprising:

machine instructions for performing the following steps:

determining that the first and second display screens are in the unfolded configuration;

receiving an input from a first finger gesture to the first display screen, wherein the first finger gesture input is for identifying a source area from which displayed data is to be copied;

receiving an input from a finger drag gesture for identifying a target area of the second display screen into which data from the source area is to be copied, wherein the target area corresponds to a location of the second display screen where the drag gesture is last detected before it ceases to be detected;

wherein the finger drag gesture includes a continuous contact with the first display screen from the first finger gesture; and

copying the data to the target area.

11. The computer readable medium of Claim 10, wherein the source area and the target area each corresponds to a displayed portion of a different application window;

wherein each of the different application windows correspond to a different software application installed on the device.

1. 12. The computer readable medium of Claim 10, wherein each of the software applications is operable on the device independently of each of the other of the software applications.

13. The computer readable medium of Claim 10, wherein the step of copying includes copying the data from the source area to a clipboard, and then automatically inputting corresponding clipboard data to the target area.

14. The computer readable medium of Claim 10, further including highlighting the source area prior to a performance of the finger drag gesture.

15. A hand-held device for performing a copy-paste operation using user finger gesture inputs, comprising:

first and second display screens of the device, wherein the device includes a folding mechanism attached to each of the first and second display screens for providing the first and seconds display screens in a folded configuration wherein the first and second display screens face in opposite directions, and in an unfolded configuration wherein the first and second display screens face in a substantially same direction;

a display manager for determining whether the first and second display screens are in an unfolded configuration or a folded configuration;

a gesture interpreter for interpreting each of a first finger gesture input to the first display screen, and a finger drag gesture input to each of the first and second display screens;

a window manager for receiving: (i) first data from the gesture interpreter, the first data indicative of the first finger gesture input, wherein the first data is for identifying a source area, displayed on the first display screen, from which displayed data is to be copied, and (ii) second data from the gesture interpreter, the second data indicative of the finger drag gesture input,

wherein the second data is for identifying a target area of the second display screen into which data from the source data is to be copied, wherein the target area corresponds to a location of the second display screen where the drag gesture is last detected by the gesture interpreter before the drag gesture ceases to be detected by the gesture interpreter;

wherein when the source area and the target area are respectively determined on the first and second display screens, data representative of a contents of the source area is copied into the target area.

16. The hand-device of Claim 15, wherein the source area is included in a display of a first application, and the target area is included in a display of a different second application.

17. The hand-device of Claim 15, wherein the first finger gesture includes pressing and holding a finger against the first display screen for a predetermined time for identifying a display location for the source area.

18. The hand-device of Claim 17, wherein the finger drag gesture includes dragging the first finger gesture on each of the first and second display screens substantially continuously across a boundary therebetween.

19. The hand-device of Claim 15 further including an application support service for inputting the data from the source area into a clipboard, and then automatically inputting corresponding clipboard data to the target area.

20. The hand-device of Claim 15, wherein the window manager highlights the source area prior to a performance of the finger drag gesture.

ABSTRACT

A multi-screen user device and methods for performing a copy-paste operation using finger gestures are disclosed. A first finger gesture is used to select a display area from which data is to be copied. Subsequently, a drag finger gesture is used to identify where the data that is to be pasted. The drag may extend across a non-display boundary between a first and second display screen of the multi-screen device.

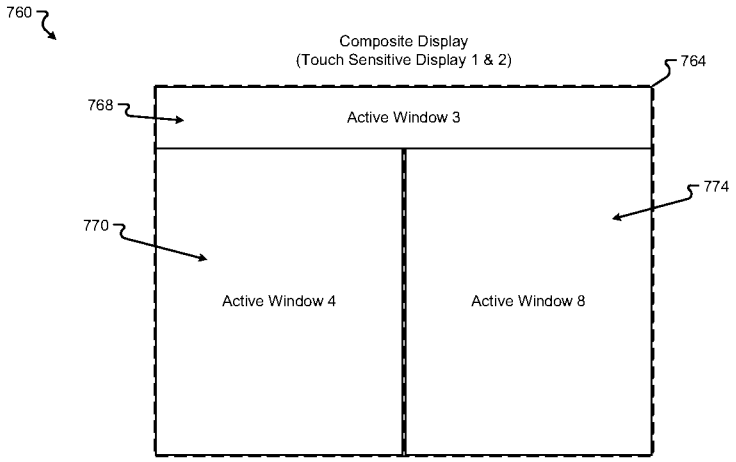


FIG. 7C

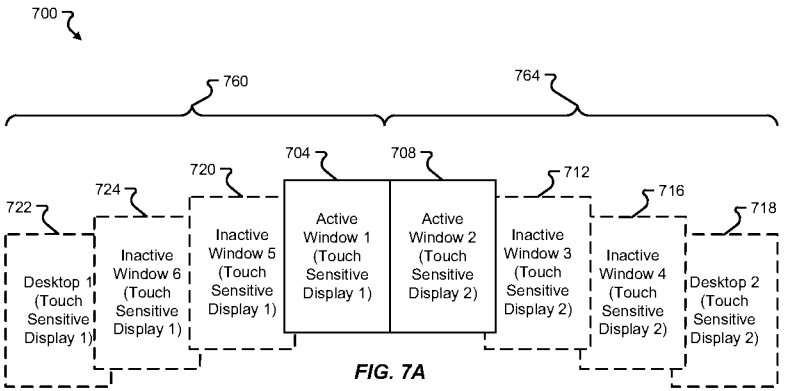


FIG. 7A

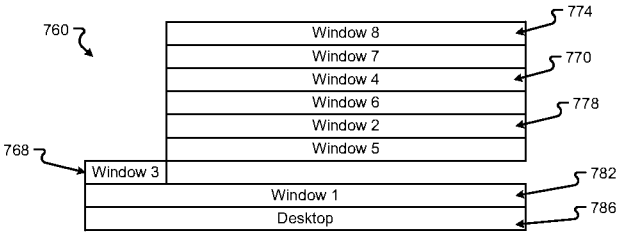


FIG. 7D

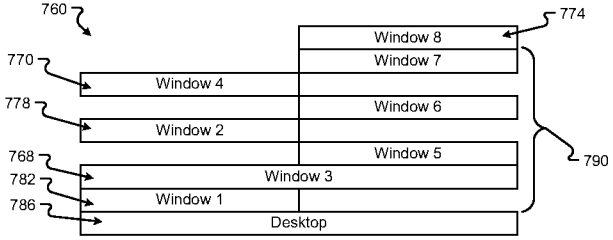


FIG. 7E

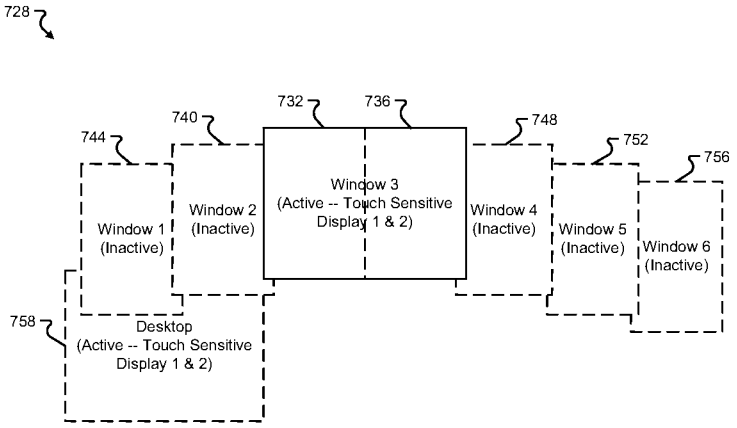


FIG. 7B

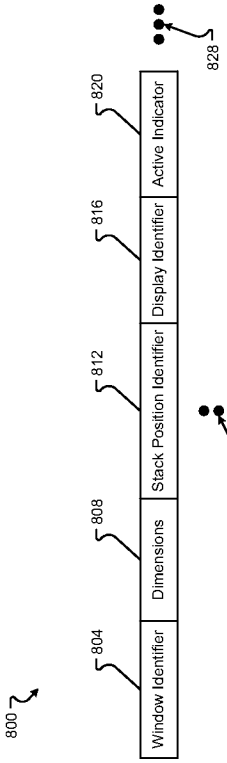


FIG. 8

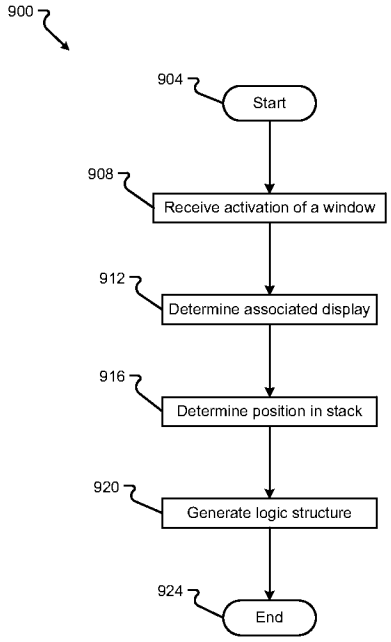


FIG. 9

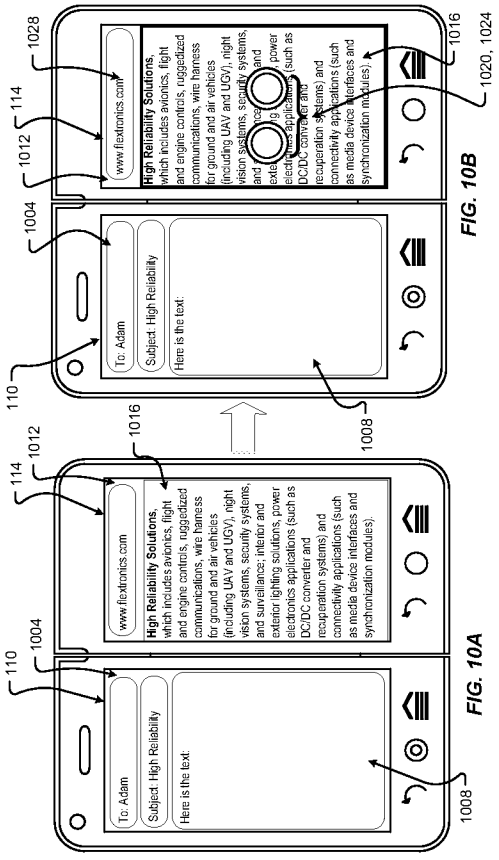


FIG. 10A

FIG. 10B

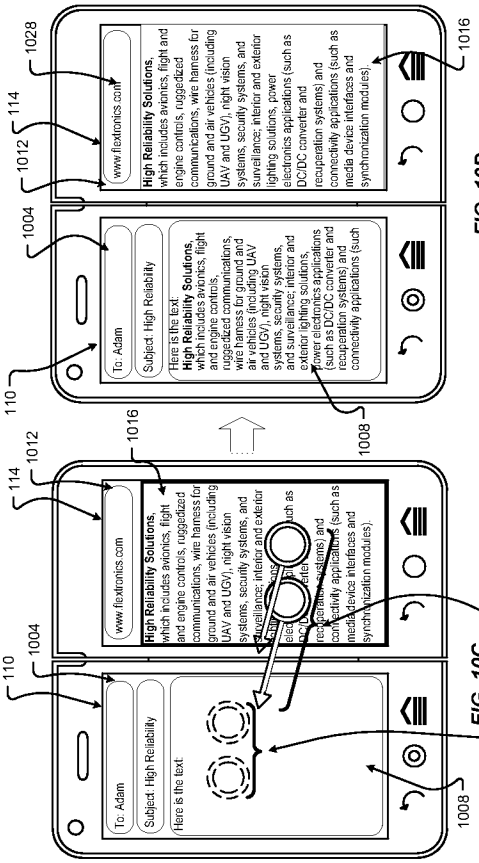


FIG. 10C

FIG. 10D

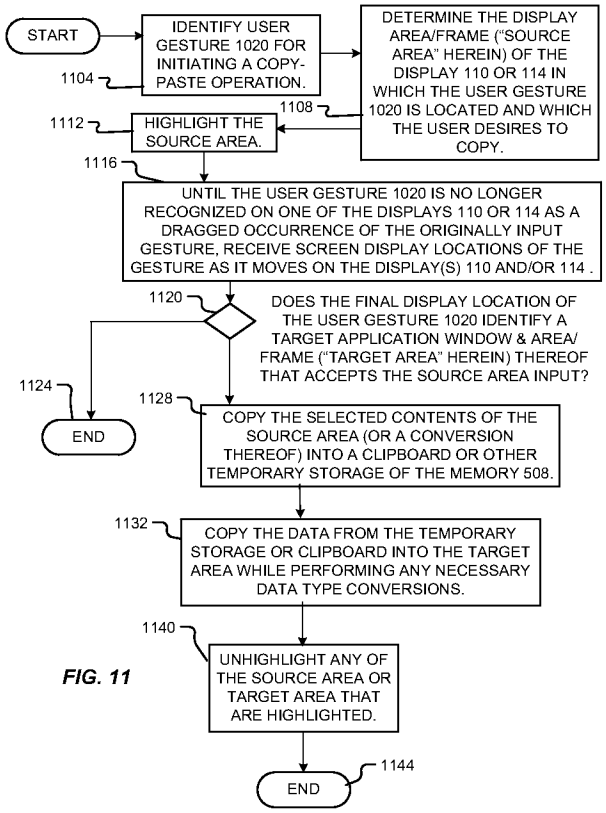

START
IDENTIFY USER GESTURE 1020 FOR INITIATING A COPY-PASTE OPERATION.
1104
DETERMINE THE DISPLAY AREA/FRAME ("SOURCE AREA" HEREIN) OF THE DISPLAY 110 OR 114 IN WHICH THE USER GESTURE 1020 IS LOCATED AND WHICH THE USER DESIRES TO COPY.
1108
1112
HIGHLIGHT THE SOURCE AREA.
1116
UNTIL THE USER GESTURE 1020 IS NO LONGER RECOGNIZED ON ONE OF THE DISPLAYS 110 OR 114 AS A DRAGGED OCCURRENCE OF THE ORIGINALLY INPUT GESTURE, RECEIVE SCREEN DISPLAY LOCATIONS OF THE GESTURE AS IT MOVES ON THE DISPLAY(S) 110 AND/OR 114 .
1120
DOES THE FINAL DISPLAY LOCATION OF THE USER GESTURE 1020 IDENTIFY A TARGET APPLICATION WINDOW & AREA/ FRAME ("TARGET AREA" HEREIN) THEREOF THAT ACCEPTS THE SOURCE AREA INPUT?
1124
END
1128
COPY THE SELECTED CONTENTS OF THE SOURCE AREA (OR A CONVERSION THEREOF) INTO A CLIPBOARD OR OTHER TEMPORARY STORAGE OF THE MEMORY 508.
1132
COPY THE DATA FROM THE TEMPORARY STORAGE OR CLIPBOARD INTO THE TARGET AREA WHILE PERFORMING ANY NECESSARY DATA TYPE CONVERSIONS.
1140
UNHIGHLIGHT ANY OF THE SOURCE AREA OR TARGET AREA THAT ARE HIGHLIGHTED.
1144
END

FIG. 11

METHOD AND SYSTEM FOR PERFORMING DRAG AND DROP OPERATIONS ON A DEVICE VIA USER GESTURES

SUMMARY

[0001] There is a need for a multi-display handheld computing device that provides for enhanced power and/or versatility compared to conventional single display handheld computing devices. In particular, using the terms described hereinabove, the present disclosure is directed to copying data from one application display location, or screen, to a display location of another such application; e.g., a drag and drop operation as this term is used in the art. More particularly, the present disclosure is directed to performing such a drag and drop operation using gestures detected on the surface of one or more touch sensitive screen displays.

[0002] In one embodiment, the method of the drag and drop operation disclosed herein includes detecting that a user is using his/her finger(s) (or other means, e.g., a stylus) to perform a gesture such as a "press and hold" gesture, wherein the user presses and holds (e.g., for at least two seconds) at a desired location on a touch sensitive screen display for identifying underneath the "press and hold" location, the information that the user desires to copy. Upon detecting this "press and hold" user gesture and the display location therefor, the device determines a displayed informational content associated with the location (more particularly, associated with an application window display area/frame referred to herein as a "source area") that is appropriate for copying. A clipboard utility may be provided by the device for copying the selected information from the source area. Subsequently, upon the user inputting additional user gestures to the touch sensitive display screen(s), e.g., by dragging his/her finger gesture configuration thereon, the user can designate to the device an application window (more particularly, an application window display area/frame therein, referred to herein as a "target area") in which the selected information is pasted therein.

[0003] For a multi-display handheld computing device, the drag and drop operation may be performed over multiple touch sensitive display screens, wherein the information from an identified source area is on a first screen display, and the user identified target area is on a second screen display. Moreover, in performing such an inter-screen drag and drop operation (via gestures), the configuration of the display screens' relative physical orientation to one another may be taken into account. In particular, since the multi-display device may allow the display screens to be folded relative to one another, or alternatively, unfolded such that each of the display screens face a same direction, such inter-screen drag and drop may occur only when the the display screens are unfolded relative to one another.

[0004] In a further embodiment, the source area (and/or data therein) may be highlighted during its selection process in order to provide the user with appropriate visual feedback that the desired informational content has been selected for the drag and drop operation.

[0005]

[0006] These and other needs are addressed by the various aspects, embodiments, and/or configurations of the present disclosure. Also, while the disclosure is presented in terms of exemplary embodiments, it should be appreciated that individual aspects of the disclosure can be separately claimed.

[0007] In one embodiment, the present disclosure describes a method for performing a drag and drop operation using user finger gesture input to a device having at least one display screen, comprising the steps (a) through (c) following:

[0008] (a) receiving a first finger gesture input to a touch sensitive display device, wherein the first finger gesture input is for identifying a source area from which displayed data is to be copied;

[0009] (b) receiving a finger drag gesture input for identifying a target area of the display into which data from the source data is to be copied, wherein the target area corresponds to a location on the display where the drag gesture is last detected before it ceases to be detected; and

[0010] (c) copying the data from the source area to the target area.

[0011] In a related embodiment, the present disclosure describes the multi-display device as having the following elements (a) through (d):

[0012] (a) first and second display screens of the device, wherein the device includes a folding mechanism attached to each of the first and second display screens for providing the first and seconds display screens in a folded configuration wherein the first and second display screens face in opposite directions, and in an unfolded configuration wherein the first and second display screens face in a substantially same direction;

[0013] (b) a multi-display manager for determining that the first and second display screens are in an unfolded configuration;

[0014] (c) a gesture interpreter for interpreting each of a first finger gesture input to the first display screen, and a finger drag gesture input to each of the first and second display screens;

[0015] (d) a window manager for receiving: (i) first data from the gesture interpreter, the first data indicative of the first finger gesture input, wherein the first data is for identifying a source area, displayed on the first display screen, from which displayed data is to be copied, and (ii) second input from the gesture interpreter, the second data indicative of the finger drag gesture input, wherein the second data is for identifying a target area of the second display screen into which data from the source data is to be copied, wherein the target area corresponds to a location of the second display screen where the drag gesture is last detected by the gesture interpreter before the drag gesture ceases to be detected by the gesture interpreter;

[0016] wherein when the source area and the target area are respectively determined on the first and second display screens, data representative of a contents of the source area is copied into the target area.

[0017]

[0018] The present disclosure can provide a number of advantages depending on the particular aspect, embodiment, and/or configuration. In particular, the above described drag and drop operation as provided via user gesture input on a multi-display device provide an more intuitive and enhanced user experience for a user interacting with an embodiment of the multi-display device. These and other advantages will be apparent from the disclosure.

[0019] The preceding is a simplified summary of the disclosure to provide an understanding of some aspects of the disclosure. This summary is neither an extensive nor exhaustive overview of the disclosure and its various aspects, embodiments, and/or configurations. It is intended neither to identify key or critical elements of the disclosure nor to delineate the scope of the disclosure but to present selected concepts of the disclosure in a simplified form as an introduction to the more detailed description presented below. As will be appreciated, other aspects, embodiments, and/or configurations of the disclosure are possible utilizing, alone or in combination, one or more of the features set forth above or described in detail below.

BRIEF DESCRIPTION OF THE DRAWINGS

[0020] FIGS. 10A through 10D show an illustration of a drag and drop operation performed via finger gestures, wherein the drag and drop operation copies from an application 564 displayed on display 114, and the paste task is into the display of another application 564 displayed on the display 110;

[0021] FIG. 11 is a flowchart providing high level steps for performed an embodiment of the drag and drop operation.

DETAILED DESCRIPTION

[0100] In at least one embodiment of the device 100, the MDM 524 (and its Gesture Module 576) may be operable for assisting the Window Management Module 532 in dragging and dropping the contexts of one application 564 to another via user gestures. In particular, to copy data from one location, or screen, to another the user presses and holds an area on the screen where the data that is desired to be copied is located; the data is then automatically highlighted and selected, then the user can drag the highlighted data to a destination location via finger gestures. This drag operation can cross over a first screen to a second screen where the data may be pasted. Figs. 10A through 10D is illustrative of the present drag and drop feature of an embodiment of the device 100. Referring to Fig. 10A, the device 100 is shown in an open portrait state, wherein the display 110 displays a email application window 1004 for composing a new email to "Adam". Note that the content area 1008 of the email currently states "Here is the text:". The display 114 shows an illustrative presentation of an Internet browser window 1012, wherein the URL for the webpage being displayed is "www.flextronics.com", and at least a portion of the contents of the webpage is shown in the browser display area/frame 1016 In Fig. 10B, a user inputs a user gesture 1020 into the display area/frame 1016 for selecting the contents thereof. In the presently illustrated embodiment, the user gesture 1020 is a two finger press and hold gesture (e.g., a two finger "long press" 424, Fig. 4B), wherein the user presses with two fingers and maintains the finger pressure on the display with both fingers for, e.g., at least two seconds for thereby selecting the contents of the display area/frame 1016 In Fig. 10B, this two finger press and hold gesture 1020 is represented by the

pair of concentric circles 1024. However, it is within the scope of the present disclosure that other/alternative user gestures may be used to select the contents of a display area/frame, e.g., 1016. For example, instead (or in addition to) the Gesture Module 576 being programmed to recognize the two finger press and hold gesture 1020 for selecting the contents of a display area/frame, the Gesture Module may be programmed to recognize a single finger press and hold combined with a finger tap 420 for selecting such contents. Note that the user could have selected the URL area/frame 1028 instead of the area/frame 1016 if the user gesture 1020 would have been applied to the URL area/frame. Further note that in order for the Window Management Module 532 to identify the particular area/frame in which the user gesture 1020 is being applied, the MDM, it may be necessary for the MDM to not only interrogate the window stack for the display (110 or 114) to which the user gesture is being input (for determining, e.g., the window in which the gesture is being input), but also interrogate the display data structure for the application window selected for determining the (any) frame within the application window to select.

[0101] Once the Gesture Module 576 recognizes the user gesture 1020 for selecting the area/frame of an application window (e.g., window 1012), the selected area/frame may be highlighted (or alternatively, only its border may be highlighted) on its display (110 or 114) so that the user receives feedback as to what portion of the screen display has been (or will be) copied into another application window. In Fig. 10B, border of the browser display area/frame 1016 is identified by a thick line width to indicate that this display area/frame has been selected by the Window Management Module 532.

[0102] Note that for hierarchical display areas/frames, the first area/frame selected by the Window Management Module 532 may be larger or smaller than what the user desires to copy. Accordingly, embodiments may allow the user to input additional gestures for selecting the desired portion of a display area/frame. For example, the user may cease the press and hold gesture with one of his/her fingers, and use this finger to quickly tap the display (110 or 114) for selecting another display area/frame within the hierarchy of display areas/frames, which would be highlighted instead of the previously selected display area/frame from the hierarchy. Thus, the user may precisely select the content to be copied into the display of another application 564.

[0103] Once the desired display area/frame is selected for copying, the user may drag his/her fingers that are pressing on the display area/frame (e.g., 1016) to be copied so that the

termination of the drag identifies the window (or display area/frame thereof) of the target application 564 in which the selected information is to be copied. Fig. 10C is illustrative of this. In particular, drag gesture 1030 is a representation of a drag of the user gesture 1020, wherein Fig. 10C further illustrates the user dragging of the user gesture 1020 from the display 114 to the drag termination location 1032 in the display 110 thereby designating to the Window Management Module 532 the target display area/frame 1008 in which to copy the information selected from display area/frame 1016. Note that the drag gesture 1030 may commence by moving the gesture 1020 across the display 100 substantially without releasing contact for touching the display 110; e.g., it may be determined that there is substantially no release of display 110 contact in transitioning from the gesture 1020 to the drag gesture 1030 if it is detected at least every 0.1 milliseconds. Moreover, in one embodiment, the selected information is copied as it resides in memory 508; e.g., text is copied as text data, pictures are copied as picture data so that data manipulations can be performed (to the degree such are supported) in the target application 564. Thus, the device 100 allows displayed information (e.g., image, text, etc.) to be selected with a simple user gesture and dragged to a new location without requiring additional user steps or lifting of the user's fingers or stylus from the displays 110 and 114. Moreover, the copied information can be copied and pasted to a different display 110 or 114 of the device by the drag user gesture 1030 extending onto both the displays 110 and 114. Note that in one embodiment, the drag user gesture may be accompanied with a corresponding drag of an image of the information to be pasted into the target application 564. In particular, the drag image may be generated as a reduced size replica of the selected information, and/or as a semi-transparent replica of the selected information wherein the user performing the drag can both recognize what is being dragged (for pasting), and identify the displayed presentations over which the drag image is being dragged.

[0104] Accordingly, Fig. 10D shows the resulting configuration of the displays 110 and 114 afrer the text of browser display area/frame 1016 is copied and pasted into the email application window 1004 (and more particularly, into the email content area 1008. Note that browser display area/frame 1016 is not longer highlighted since the drag and drop operation has been completed.

[0105] The flowchart of Fig. 11 shows illustrative high level steps that may be performed by an embodiment of the device 100 in performing a drag and drop operation as described

hereinabove with reference to Figs. 10A-10D. However, before describing the flowchart steps, it noteworthy that in order for the drag and drop operation to be eligible for being performed wherein the user drag gesture 1020 extends from one of the displays 110, 114 to the other of these displays, the device 100 must in either the open portrait or open landscape state as determined by the MDM 524. Otherwise, the drag and drop operation is confined to a single one of displays 110 or 114. Accordingly, in step 1104, the Gesture Module 576 determines that a user input to one of the displays 110 or 114 is a user gesture (e.g., user gesture 1020) for at least selecting a display area/frame of an application window. In step 1108, upon receiving input from the Gesture Module 576 indicating that such a user gesture has been detected as occurring on one of the displays 110 or 114, the Window Management Module 532 determines a display area/frame ("source area" herein) of the window application where the user gesture was first detected, wherein the source area is determined by the user as the desired area/frame to copy. In performing the step 1108, the Window Management Module 532 may, in one embodiment, receive additional inputs from the Gesture Module 576 for notifying the Window Management Module 532 of additional user gesture input for identifying various display areas/frames in a display hierarchy of such display areas/frames. For example, the user may input the user gesture 1020 upon a displayed picture on a webpage, and the webpage may be initially identified by the Window Management Module 532. Thus, the Window Management Module 532 may cause the webpage (or a frame boundary thereof) to be highlighted to the user thereby indicating that the webpage has been selected for coping. Accordingly, the user may provide an additional user gesture (e.g. a finger tap 400 as described above with reference to Fig. 10B) to indicate that another area/frame of the display is desired to be copied. Accordingly, assuming there is no intermediate display area/frame in a hierarchy that ends in the picture, upon the Window Management Module 532 receiving data from the Gesture Module 576 indicating that the additional user gesture has been input by the user, the MDM selects the picture, and highlights this picture on the display (110 or 114) to confirm to the user the information selected to be copied.

[0106] Thus, as indicated in the example of selecting a displayed picture above, the Window Management Module 532 may perform the step 1112 for highlighting the display area/frame that has been selected for copying. Following this step, upon receiving additional input from the Gesture Module 576 regarding (any) movement (or other change) in the user gesture 1020, in

step 1116, the Window Management Module 532 receives and stores a display location for each such additional input. Note that previous display locations corresponding to a previous location of the user gesture 1020 may be overwritten with each new display location. Moreover, the Window Management Module 532 may be programmed to create, e.g., a semi-transparent copy (or a non-transparent iconic representation) of the selected information which may be displayed as following beneath the dragging of the user gesture 1020 over the displays 110 and/or 114. Furthermore, for each input the Window Management Module 532 receives of the user gesture 1020 display location, the MDM 524 may in real time: (i) determine a (any) display area/frame identified by the user gesture display location, and once determined, (ii) highlight or otherwise identify on the display(s) 110 and/or 114 such a determined display area/frame (or the border thereof) as being a candidate display area/frame that can accept the selected information in a paste operation. Note that in determining (i) immediately above, if the user gesture display location identifies a hierarchical collection of display areas/frames, then the Window Management Module 532 may determine the lowest ordered (e.g., smallest) display area/frame in the hierarchy that can accept the selected information. Thus, for example, if the content area 1008 of Figs. 10 included a form with entries that can accept user input, then if a first entry was a check box, then if the user gesture 1020 were dragged over this first entry, it would not be identified as appropriate for receiving the selected information, and instead, a higher ordered display area/frame would be determined, e.g., the content area 1008. If, however, a second form entry is determined to be beneath the current location of the user gesture 1020, and this second form entry is able to accept the selected information for the drag and drop operation (e.g., textual webpage input as selected for the drag and drop operation in Fig. 10B), then the area/frame for this second form entry would be identified (e.g., via highlighting) on its display(s) 110 or 114 as a candidate for receiving a paste of the selected information.

[0107] Assuming the MDM 524 has informed the Gesture Module 576 that the device 100 is in an open (unfolded) configuration, then when the user gesture 1020 drags off its display 110, 114, the Gesture Module 576 may attempt to correlate a corresponding drag gesture 1020 on the other one of the displays 110, 114. In particular, assuming the device 100 is unfolded (the Gesture Module 576 has been informed of such), the Gesture Module 576 may detect that the gesture 1020 has lefr its display on a side between the displays 110, 114 (rather than some other side which may be interpreted as a termination of the drag and drop operation). Accordingly, the

Gesture Module 576 may wait a predetermined time of, e.g., at most 0.5 seconds for detecting a corresponding drag gesture 1020 near the border of the other of the displays 110, 114, wherein this border is closest to the border of the display 110, 114 from which the drag gesture 1020 ceased to be detected. Thus, the device 100 can track the user drag gesture 1020 from one of the displays 110, 114 to the other of the displays, wherein the drag gesture 1020 substantially continuously traverses from one of the displays 110, 114 to the other display.

[0108] In step 1120, upon the Window Management Module 532 receiving input from the Gesture Module 576 that the user gesture 1020 has ceased to be detected on one of the displays 110 or 114, the MDM determines whether there is an area/frame underneath the last stored display location of the user gesture 1020, wherein this area/frame (the "target area" herein) accepts input as provided by the source area. If not, then there is no appropriate area/frame in which to paste the selected information, and accordingly the process corresponding to flowchart of Fig. 11 terminates at 1124 without performing the drag and drop operation. Note that a user can use this technique to safely terminate the drag and drop operation without causing a paste of the selected information. That is, the user could terminate the drag of the user gesture 1030 over a display location that does not identify an area/frame of the displays 110 or 114 that accepts the selected information of the source area as input. However, in an additional/alternative embodiment, a user may terminate the drag and drop operation without completing the paste by accelerating the user gesture 1030 off the displays 110 and 114.

[0109] If, on-the-other-hand, step 1120 results in identifying a display area/frame of an application window that can accept the source area input, then in step 1128, the selected contents of source area is copied to a temporary data storage (not separately shown) of memory 508 (e.g., a "clipboard" as one skilled in the art will understand.).

[0110] Subsequently, in step 1132, this source area information that is newly entered in the temporary data storage is then automatically (e.g., without requiring user input) copied into the target area by the Window Manager Module 532. Note that for applications 564 in which, e.g., text, tables, pictures, videos, and/or combinations thereof may be input, such applications are likely to be configured when installed in the device 100 so that these applications can accept input from a clipboard as part of the device's application support services provided by the application manager 562. Moreover, note that the paste of the source area information into the target may require data type conversions so that an appropriate version of the selected content of

the source area can be pasted into the target area. For example, if the selected content of the source area is Microsoft Word® text, and the target area the Google Chrome® then the selected content may need to be converted to a "rich text" format as one skilled in the art will understand.

[0111] Subsequently, in step 1140, if there are highlighted portions of the displays 110,114 due to the present drag and drop operation, such highlighting is removed or turned off by the Window Management Module 532. In particular, the Window Management Module 532 may turn off any highlighting of the source and target areas. Subsequently, the flowchart of Fig. 11 terminates at 1144.

What is claimed is:

1. A method for performing a drag and drop operation using user finger gesture input to a device having a plurality of touch sensitive display, comprising:

receiving an input of a first finger gesture to a first of the touch sensitive displays, wherein the first finger gesture input is for identifying a source area of the first touch sensitive display, wherein the source area includes data to be copied;

receiving an input of a finger drag gesture for identifying a target area of a second of the touch sensitive displays into which the data from the source data is to be copied, wherein the finger drag gesture extends across a boundary between the first touch sensitive display and the second touch sensitive display, wherein the first and second touch sensitive displays are foldable relative to one another along the boundary;

wherein the target area corresponds to a location on the second touch sensitive display where the drag gesture is last detected before it ceases to be detected;

changing a display of the target area for identifying the target area to a user as able to receive the data from the source area; and

copying the data into the target area.

2. The method of Claim 1, wherein at least one of the first and second touch sensitive displays includes a capacitive touch sensitive area for receiving the finger gesture input.

3. The method of Claim 1, further including a step of outputting on the first touch sensitive display a display change for identifying the source area to the user.

4. The method of Claim 3, further including a step of selecting the target area identified by the location on the display where the drag gesture is last detected before it ceases to be detected.

5. The method of Claim 4, wherein the step of selecting includes selecting a display area that accepts a clipboard input, wherein the data from the source area is copied to the clipboard, and copied from the clipboard to the target area.

6. The method of Claim 1, further including selecting the source area by identifying a location on the first touch sensitive display for the first finger gesture input, and changing a display presentation of the source area.

7. The method of Claim 1, wherein the first finger gesture includes pressing and holding two fingers against the display for a predetermined time for identifying a display location of the source area.

8. The method of Claim 7, wherein the finger drag gesture includes dragging the finger gesture substantially continuously from the display location to the location on the second touch sensitive display on the touch sensitive display device.

9. The method of Claim 1, further including identifying to the user one or more display areas as the user inputs a portion of the finger drag gesture over each of the one or more display areas, wherein each of the one or more display areas are determined to be acceptable for pasting the data therein.

10. A computer readable medium for performing a drag and drop operation using user finger gesture inputs to first and second display screens of a device, wherein the device includes a folding mechanism attached to each of the first and second display screens for providing the first and seconds display screens in a folded configuration wherein the first and second display screens face in opposite directions, and in an unfolded configuration wherein the first and second display screens face in a substantially same direction, comprising:

machine instructions for performing the following steps:

determining that the first and second display screens are in the unfolded configuration;

receiving an input of a first finger gesture to the first display screen, wherein the first finger gesture input is for identifying a source area of the first display screen, wherein the source area includes data to be copied;

receiving an input of a finger drag gesture for identifying a target area of the second display screen into which the data from the source data is to be copied, wherein the finger drag

gesture extends across a boundary between the first display screen and the second display screen, wherein the first and second display screens are foldable relative to one another along the boundary;

wherein the target area corresponds to a location on the second display screen where the drag gesture is last detected before it ceases to be detected;

changing a display of the target area for identifying the target area to a user as able to receive the data from the source area; and

copying the data into the target area.

11. The computer readable medium of Claim 10, wherein the source area and the target area each correspond to a displayed portion of a different application window;

wherein each of the different application windows corresponds to a different software application installed on the device.

12. The computer readable medium of Claim 10, wherein each of the software applications is operable on the device independently of each of the other of the software applications.

13. The computer readable medium of Claim 10, wherein the step of copying includes copying the data from the source area to a clipboard, and then automatically inputting corresponding clipboard data to the target area.

14. The computer readable medium of Claim 10, further including highlighting the source area prior to a performance of the finger drag gesture.

15. A hand-held device for performing a drag and drop operation using user gestures as inputs, comprising:

first and second display screens of the device, wherein the device includes a folding mechanism attached to each of the first and second display screens for providing the first and seconds display screens in a folded configuration wherein the first and second display screens

face in opposite directions, and in an unfolded configuration wherein the first and second display screens face in a substantially same direction;

a display manager for determining whether the first and second display screens are in an unfolded configuration or a folded configuration;

a gesture interpreter for interpreting an input of a first gesture to the first display screen for identifying a first area, and subsequently interpreting an input of a second gesture for selecting a second area, wherein the first area includes the second area;

a window manager for receiving: (i) first data from the gesture interpreter, the first data includes location data indicative of a location on the first display screen of the first gesture input, and (ii) second data from the gesture interpreter, the second data for identifying the second area;

wherein the window manager uses the location data for determining the first area, and changing a display presentation of the first area for identifying it to a user;

wherein afrer the change in display presentation of the first area, the gesture interpreter receives the second gesture input, and the window manager uses the second data to change a display presentation of the second area for identifying the second area to the user as a source area for obtaining data for the drag and drop operation;

wherein when the window manager receives input from the gesture interpreter of a drag gesture being input for performing a drag for the source area, a corresponding image to be dragged is generated;

wherein the window manager uses a display screen location from the drag gesture for identifying a target area of the second display screen into which data from the source data is to be copied, wherein the target area corresponds to a location of the second display screen where the drag gesture is last detected by the gesture interpreter before the drag gesture ceases to be detected by the gesture interpreter;

wherein the data from the source area is copied into the target area.

16. The hand-device of Claim 15, wherein the source area is included in a display of a first application, and the target area is included in a display of a different second application.

17. The hand-device of Claim 15, wherein the first gesture includes pressing and holding a finger against the first display screen for a predetermined time for identifying a display location for the first area.

18. The hand-device of Claim 17, wherein the drag gesture includes dragging the first gesture on each of the first and second display screens substantially continuously across a boundary therebetween.

19. The hand-device of Claim 15 further including an application support service for inputting the data from the source area into a clipboard, and then automatically inputting corresponding clipboard data to the target area.

20. The hand-device of Claim 15, wherein the window manager highlights the source area prior to a performance of the drag gesture.

ABSTRACT

A multi-screen user device and methods for performing a drag and drop operation using finger gestures are disclosed. A first finger gesture is used to select a display area from which data is to be copied. Subsequently, a drag finger gesture is used to identify where the data that is to be pasted. The drag may extend across a non-display boundary between a first and second display screen of the multi-screen device.

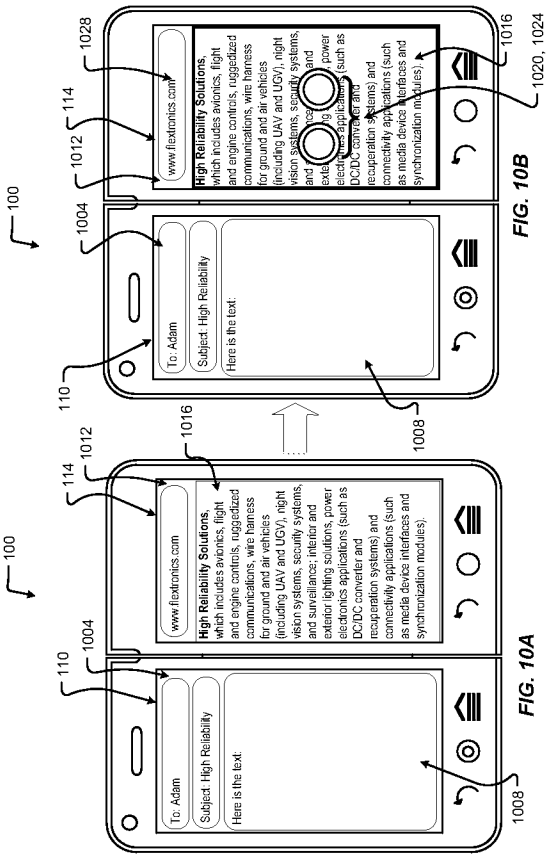


FIG. 10A

FIG. 10B

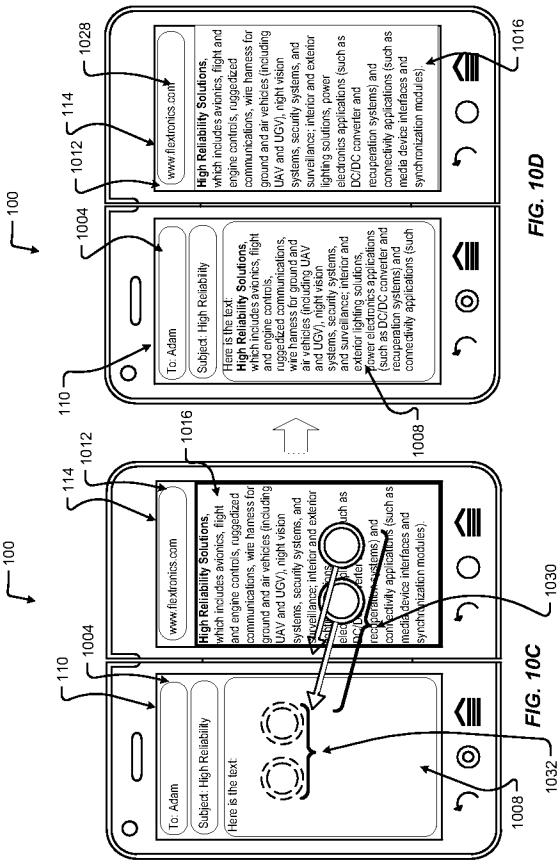


FIG. 10C

FIG. 10D

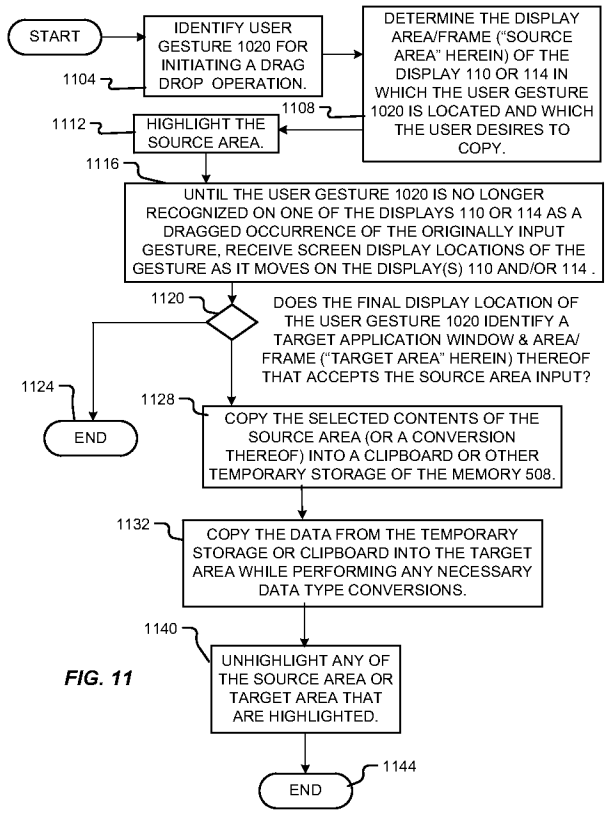


FIG. 11

UNIVERSAL CLIPBOARD

[0001] Universal Clipboard:

[0002] Figs. 7A and 7B depict two instances of data displayed by a device 100. Specifically, Fig. 7A depicts a first screen-shot of a device 100 in an open portrait state 320. While in this state, the primary screen 104 may depict data from one or more applications (e.g., a first application 704a and a second application 704b) according to locally-maintained presentation rules for the applications. The secondary screen 108 may depict another application in the form of a universal clipboard application 708. The data from the applications 704a, 704b, 708 may be rendered on the display portions 110, 114 of the screens 104, 108, respectively, or any other component of the screens 104, 108.

[0003] In some embodiments, the universal clipboard application 708 corresponds to one specific type of an application that is stored in the application store 560. The universal clipboard application 708 may be configured to store copied content from a plurality of different applications (e.g., word-processing applications, email applications, web-browsing applications, spreadsheet applications, media player applications, data sharing applications, file-navigation applications, operating systems, etc.) each of which may be formatted differently and, therefore, may treat similar media (e.g., text, audio, video, images, etc.) differently. As some non-limiting examples, the universal clipboard application 708 may be configured to store content from the first application 704a in the form of text content, hyperlinks (e.g., text representations of a URL or pointer), images, file path names, contact information, and/or video content (e.g., the image and/or audio portions thereof). The universal clipboard application 708 may also be configured to store content from the second application 704b in the form of text content, file content, etc. Since the first and second applications 704a, 704b are different, similar content-types may be formatted differently. For example, the text content from the first application 704a may be in a rich text format (RTF), unformatted text, or an object. Text content from the second application 704b, however, may be in a different format such as HTML, Unformatted Unicode Text, etc.

[0004] Each element that is copied from the applications 704a, 704b may be stored in their native format (e.g., the format in which the content exists in the application) or in another predetermined format that is universally-pasteable into different applications (e.g., image formats such as .png, .tiff, JPEG, BMP, etc.; text formats such as Unformatted Unicode Text, or plain text; or audio formats such as MPEG, WAV, AIFF, AU, MP3, etc.). The universal clipboard application 708 may also render an item display 712a-N for each element that is copied from an application. The number of items 712a-N displayed by the universal clipboard application 708 may be infinite or limited to a predetermined number of items (e.g., as defined by user preferences or default settings).

[0005] In some embodiments, the item display 712 in the universal clipboard 708 for copied content may correspond to an image, icon, text describing the copied content, or combinations thereof. Specifically, the universal clipboard can be accessed as an application where a visual representation of the copied information is displayed and can subsequently be selected by the user to be pasted.

[0006] As can be seen in fig. 7B, a third application 704c may be opened and displayed on the primary screen 104 (or whichever screen is opposite the screen displaying the universal clipboard application 708). Alternatively, or in addition, the third application 704c may be displayed on the same screen as the universal clipboard application 708, although such an embodiment is not depicted. The third application 704c can be the same as or different from the first application 704a or second application 704b.

[0007] The user of the device 100 can engage the universal clipboard application 708 by, for example, providing one or more gestures to the screen on which the universal clipboard application 708 is displayed and cause one or more copied elements to be pasted to the third application 704c as one or more pasted items 716a, 716b, etc. This can be accomplished without actually engaging the screen on which the third application 704c is displayed if the third application 704c is displayed on a screen other than the screen on which the universal clipboard application 708 is displayed. It is possible, however, that the user can engage both screens 104, 108 when executing a paste command for one or more items 712a-N on the clipboard.

[0008] In some embodiments, the gestures used to execute a paste command may be provided on the touch-sensitive display area 110, 114 of the screens 104, 108. However, the gestures used to execute a paste command may also be provided on the gesture capture region 120, 124, or the configurable area 112, 116 of the screens 104, 108. As one non-limiting example, a user may provide an input to the universal clipboard application 708 that includes a series of gestures where a first gesture in the series of gestures corresponds to a selection of the desired item 712 to be copied (e.g., as a touch, tap, or long-press gesture) and where a second gesture in the series of gestures corresponds to a movement-based gesture (e.g., a flick, drag, pinch, or spread gesture).

In some embodiments, the selection gesture may be received in the touch-sensitive display area 110, 114 that is depicting the item 712 and the movement-based gesture may received in the gesture capture region 120, 124 that is above the display of the universal clipboard application 708.

[0009] Most single or dual-screen devices will allow a user to copy content from one location to another by first selecting an image or text, then accessing a copy/paste menu pop-up window, then selecting the copy command, at which point the device stores the information to a virtual clipboard. To paste the copied content, the user must navigate to the destination location (e.g., the application at which the content will be copied), select the area where the text should be placed, access a copy/paste menu pop-up window, and then select the paste command. This copy/paste routine must be completed for each different piece of information to be copied, which requires substantial time and input from the user.

[0010] The device 100 allows a user to copy multiple contents to the universal clipboard application 708. From this universal clipboard application, a user can choose an item corresponding to already-copied content, and (while still providing inputs at the universal clipboard application 708) execute a paste command. By providing multiple screens where one screen displays the universal clipboard application and another screen displays an application, a user is enabled to easily and quickly copy and paste content between various different applications on different screens 104, 108 of the device 100. Moreover, the ability to quickly and easily copy and paste different information from one central location, the universal clipboard application 708, saves the user time and input. Thus, the device 100 operates very intuitively for the user.

[0011] Referring now to Fig. 8, a copy and paste method 800 will be described in accordance with at least some embodiments of the present disclosure. The method 800 is initiated at step 804. Specifically, a user may be interacting with one or more applications 704 on the device 100 while the device 100 is in an open state (e.g., either open portrait or open landscape) such that both screens 104, 108 are active and displaying data.

[0012] The method continues with the universal clipboard application 708 determining whether an input or series of inputs has been received that correspond to a copy command (step 808). The input or series of inputs that may correspond to a copy command can vary and may be user-configurable. As a non-limiting example, the input or series of inputs may include a

selection of content to be copied (e.g., via a selection input, a selection input coupled with a drag, a multi-finger selection input, a multi-finger drag input, etc.) and/or a selection of a paste command (e.g., via another selection input, a selection input coupled with a flick, a two-fingered flick, etc.). Advantageously, a single gesture such as a two-fingered flick can be used to execute the copy function as long as the flick is directed toward the screen displaying the universal clipboard application 708 (which may be across screens or on the same screen) and the device 100 is in an open state.

[0013] If a copy command has been received, then the selected content is copied to the universal clipboard (step 812) and the universal clipboard renders an item 712 display that corresponds to the copied content (step 816).

[0014] Thereafter, or if no copy command is received in step 808, then the method proceeds with the universal clipboard application 708 determining whether an input or series of inputs has been received that correspond to a paste command (step 820). Similar to the copy command, the paste command may comprise a single gesture received on the screen that is displaying the universal clipboard application 708. Alternatively, the paste command may comprise a series of gestures that includes a selection gesture and a motion-based gesture. All of the gestures for the paste command may be received on the screen that is displaying the universal clipboard application 708, although this is not necessary. For example, a selection gesture may be received for one of the items 712 being displayed by the universal clipboard application 708 and a movement-based gesture may be received in the gesture capture region 120, 124 of either screen 104 or screen 108, irrespective of where the universal clipboard application 708 is being displayed.

[0015] If no paste command is received, then the method returns to step 808. If, however, a paste command is received, then the content corresponding to the selected item 712 is formatted for the destination application (if it wasn't already formatted in a universally-pasteable format prior to being copied and stored by the universal clipboard application 708) (step 824) and then the content corresponding to the selected item 712 is pasted to the selected application (step 828). In some embodiments, the selected application may correspond to an application that is being displayed simultaneously with the universal clipboard application 708.

[0016] The exemplary systems and methods of this disclosure have been described in relation to display control mechanisms. However, to avoid unnecessarily obscuring the present

disclosure, the preceding description omits a number of known structures and devices. This omission is not to be construed as a limitation of the scopes of the claims. Specific details are set forth to provide an understanding of the present disclosure. It should however be appreciated that the present disclosure may be practiced in a variety of ways beyond the specific detail set forth herein.

[0017] Furthermore, while the exemplary aspects, embodiments, and/or configurations illustrated herein show the various components of the system collocated, certain components of the system can be located remotely, at distant portions of a distributed network, such as a LAN and/or the Internet, or within a dedicated system. Thus, it should be appreciated, that the components of the system can be combined in to one or more devices, such as a computer, laptop, netbook, smart phone, mobile device, etc., or collocated on a particular node of a distributed network, such as an analog and/or digital telecommunications network, a packet-switch network, or a circuit-switched network. It will be appreciated from the preceding description, and for reasons of computational efficiency, that the components of the system can be arranged at any location within a distributed network of components without affecting the operation of the system. For example, the various components can be located in a switch such as a PBX and media server, gateway, in one or more communications devices, at one or more users' premises, or some combination thereof. Similarly, one or more functional portions of the system could be distributed between a telecommunications device(s) and an associated computing device.

[0018] Furthermore, it should be appreciated that the various links connecting the elements can be wired or wireless links, or any combination thereof, or any other known or later developed element(s) that is capable of supplying and/or communicating data to and from the connected elements. These wired or wireless links can also be secure links and may be capable of communicating encrypted information. Transmission media used as links, for example, can be any suitable carrier for electrical signals, including coaxial cables, copper wire and fiber optics, and may take the form of acoustic or light waves, such as those generated during radio-wave and infra-red data communications.

[0019] Also, while the flowcharts have been discussed and illustrated in relation to a particular sequence of events, it should be appreciated that changes, additions, and omissions to this

sequence can occur without materially affecting the operation of the disclosed embodiments, configuration, and aspects.

[0020] A number of variations and modifications of the disclosure can be used. It would be possible to provide for some features of the disclosure without providing others.

[0021] In yet another embodiment, the systems and methods of this disclosure can be implemented in conjunction with a special purpose computer, a programmed microprocessor or microcontroller and peripheral integrated circuit element(s), an ASIC or other integrated circuit, a digital signal processor, a hard-wired electronic or logic circuit such as discrete element circuit, a programmable logic device or gate array such as PLD, PLA, FPGA, PAL, special purpose computer, any comparable means, or the like. In general, any device(s) or means capable of implementing the methodology illustrated herein can be used to implement the various aspects of this disclosure. Exemplary hardware that can be used for the disclosed embodiments, configurations and aspects includes computers, handheld devices, telephones (e.g., cellular, Internet enabled, digital, analog, hybrids, and others), and other hardware known in the art. Some of these devices include processors (e.g., a single or multiple microprocessors), memory, nonvolatile storage, input devices, and output devices. Furthermore, alternative software implementations including, but not limited to, distributed processing or component/object distributed processing, parallel processing, or virtual machine processing can also be constructed to implement the methods described herein.

[0022] In yet another embodiment, the disclosed methods may be readily implemented in conjunction with software using object or object-oriented software development environments that provide portable source code that can be used on a variety of computer or workstation platforms. Alternatively, the disclosed system may be implemented partially or fully in hardware using standard logic circuits or VLSI design. Whether software or hardware is used to implement the systems in accordance with this disclosure is dependent on the speed and/or efficiency requirements of the system, the particular function, and the particular software or hardware systems or microprocessor or microcomputer systems being utilized.

[0023] In yet another embodiment, the disclosed methods may be partially implemented in software that can be stored on a storage medium, executed on programmed general-purpose computer with the cooperation of a controller and memory, a special purpose computer, a microprocessor, or the like. In these instances, the systems and methods of this disclosure can be

implemented as program embedded on personal computer such as an applet, JAVA® or CGI script, as a resource residing on a server or computer workstation, as a routine embedded in a dedicated measurement system, system component, or the like. The system can also be implemented by physically incorporating the system and/or method into a software and/or hardware system.

[0024] Although the present disclosure describes components and functions implemented in the aspects, embodiments, and/or configurations with reference to particular standards and protocols, the aspects, embodiments, and/or configurations are not limited to such standards and protocols. Other similar standards and protocols not mentioned herein are in existence and are considered to be included in the present disclosure. Moreover, the standards and protocols mentioned herein and other similar standards and protocols not mentioned herein are periodically superseded by faster or more effective equivalents having essentially the same functions. Such replacement standards and protocols having the same functions are considered equivalents included in the present disclosure.

[0025] The present disclosure, in various aspects, embodiments, and/or configurations, includes components, methods, processes, systems and/or apparatus substantially as depicted and described herein, including various aspects, embodiments, configurations embodiments, subcombinations, and/or subsets thereof. Those of skill in the art will understand how to make and use the disclosed aspects, embodiments, and/or configurations afrer understanding the present disclosure. The present disclosure, in various aspects, embodiments, and/or configurations, includes providing devices and processes in the absence of items not depicted and/or described herein or in various aspects, embodiments, and/or configurations hereof, including in the absence of such items as may have been used in previous devices or processes, e.g., for improving performance, achieving ease and\or reducing cost of implementation.

[0026] The foregoing discussion has been presented for purposes of illustration and description. The foregoing is not intended to limit the disclosure to the form or forms disclosed herein. In the foregoing Detailed Description for example, various features of the disclosure are grouped together in one or more aspects, embodiments, and/or configurations for the purpose of streamlining the disclosure. The features of the aspects, embodiments, and/or configurations of the disclosure may be combined in alternate aspects, embodiments, and/or configurations other than those discussed above. This method of disclosure is not to be interpreted as reflecting an

intention that the claims require more features than are expressly recited in each claim. Rather, as the following claims reflect, inventive aspects lie in less than all features of a single foregoing disclosed aspect, embodiment, and/or configuration. Thus, the following claims are hereby incorporated into this Detailed Description, with each claim standing on its own as a separate preferred embodiment of the disclosure.

[0027] Moreover, though the description has included description of one or more aspects, embodiments, and/or configurations and certain variations and modifications, other variations, combinations, and modifications are within the scope of the disclosure, e.g., as may be within the skill and knowledge of those in the art, afrer understanding the present disclosure. It is intended to obtain rights which include alternative aspects, embodiments, and/or configurations to the extent permitted, including alternate, interchangeable and/or equivalent structures, functions, ranges or steps to those claimed, whether or not such alternate, interchangeable and/or equivalent structures, functions, ranges or steps are disclosed herein, and without intending to publicly dedicate any patentable subject matter.

What is claimed is:

1. A method of operating a multi-screen device, comprising:

receiving a first input corresponding to an instruction to copy first selected content from a first application;

copying the first selected content;

rendering a display of a first item with a universal clipboard application, the first item corresponding to the first selected content;

receiving, on a second screen of the multi-screen device, a second input corresponding to a selection of the first item and an instruction to paste the first item in a second application that is different from the first application; and

pasting the first selected content to the second application on a first screen of the multi-screen device.

2. The method of claim 1, wherein the second input is received entirely on the second screen.

3. The method of claim 2, wherein the second input comprises a flick gesture.

4. The method of claim 2, wherein the second input is a series of gestures including at least one of a touch gesture, tap gesture, and long press gesture as well as at least one of a drag gesture, a flick gesture, a pinch gesture, and a spread gesture.

5. The method of claim 4, wherein the at least one of a touch gesture, tap gesture, and long press gesture is received on a touch sensitive display area of the second screen that is displaying the first item and wherein the at least one of a drag gesture, flick gesture, pinch gesture, and spread gesture is received on a gesture capture region of the second screen.

6. The method of claim 1, wherein the universal clipboard application is displayed on the second screen of the multi-screen device simultaneously with the second application being displayed on the first screen of the multi-screen device.

7. The method of claim 1, wherein the first application is displayed on the first screen of the multi-screen device, wherein the first input is received on the first screen of the multi-screen device, wherein the first input comprises at least one of a drag and flick gesture, and

wherein the universal clipboard is displayed on the second screen of the multi-screen device when the first input is received.

8. The method of claim 7, wherein the first input is only enabled to correspond to the instruction to copy the first selected content when the multi-screen device is in an open state.

9. The method of claim 1, wherein the first selected content comprises at least one of text, an image, a video, a file path, and a file.

10. A non-transitory computer-readable medium having stored thereon instructions that cause a computing system to execute a method, the instructions comprising:

instructions configured to receive a first input corresponding to an instruction to copy first selected content from a first application;

instructions configured to copy the first selected content;

instructions configured to render a display of a first item with a universal clipboard application, the first item corresponding to the first selected content;

instructions configured to determine that a second input is received on a second screen of the multi-screen device, the second input corresponding to a selection of the first item and an instruction to paste the first item in a second application that is different from the first application; and

instructions configured to paste the first selected content to the second application on a first screen of the multi-screen device.

11. The computer-readable medium of claim 10, wherein the second input is received entirely on the second screen.

12. The computer-readable medium of claim 11, wherein the second input comprises a drag gesture.

13. The computer-readable medium of claim 11, wherein the second input is a series of gestures including at least one of a touch gesture, tap gesture, and long press gesture as well as at least one of a drag gesture, a flick gesture, a pinch gesture, and a spread gesture.

14. The computer-readable medium of claim 13, wherein the at least one of a touch gesture, tap gesture, and long press gesture is received on a touch sensitive display area of the second screen that is displaying the first item and wherein the at least one of a drag gesture, flick gesture, pinch gesture, and spread gesture is received on a gesture capture region of the second screen.

15. The computer-readable medium of claim 10, wherein the universal clipboard application is displayed on the second screen of the multi-screen device simultaneously with the second application being displayed on the first screen of the multi-screen device.

16. The computer-readable medium of claim 10, wherein the first application is displayed on the first screen of the multi-screen device, wherein the first input is received on the first screen of the multi-screen device, wherein the first input comprises at least one of a drag and flick gesture, and wherein the universal clipboard is displayed on the second screen of the multi-screen device when the first input is received.

17. A multi-screen user device, comprising:

a first screen including a first display area;

a second screen including a second display area; and

a computer-readable medium having instructions stored thereon that include:

a universal clipboard application configured to receive a first input corresponding to an instruction to copy first selected content from a first application, copy the first selected content, render a display of a first item corresponding to the first selected content, determine that a second input is received on a second screen of the multi-screen device, the second input corresponding to a selection of the first item and an instruction to paste the first item in a second application that is different from the first application, and in response to receiving the second input, paste the first selected content to the second application on a first screen of the multi-screen device.

18. The device of claim 17, wherein the second input is received entirely on the second screen.

19. The device of claim 18, wherein the second input comprises at least one of a flick gesture and a drag gesture.

20. The device of claim 17, wherein the universal clipboard application is displayed on the second screen of the multi-screen device simultaneously with the second application being displayed on the first screen of the multi-screen device.

Attorney Docket No. 6583-75

ABSTRACT

A multi-screen user device and a universal clipboard application are described. Specifically, the universal clipboard application can be open on one screen of the multi-screen device while another application, from which data is either being copied to the clipboard application or pasted from the clipboard application, is open on another screen of the multi-screen device. Inputs received on the screen displaying the universal clipboard application can cause content to be copied to an application being displayed on the other screen of the multi-screen device, thereby providing an intuitive user interface for the universal clipboard application.

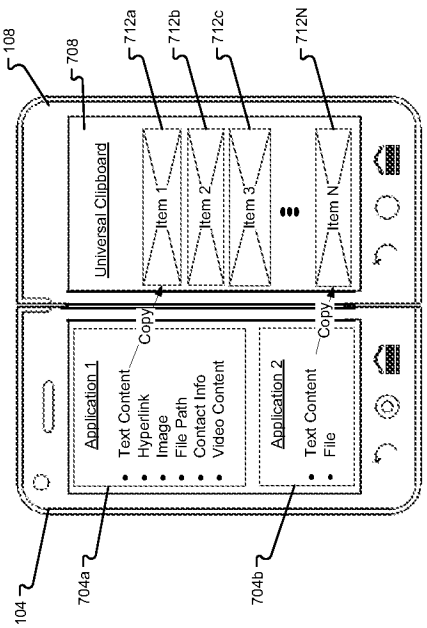


Fig. 7A

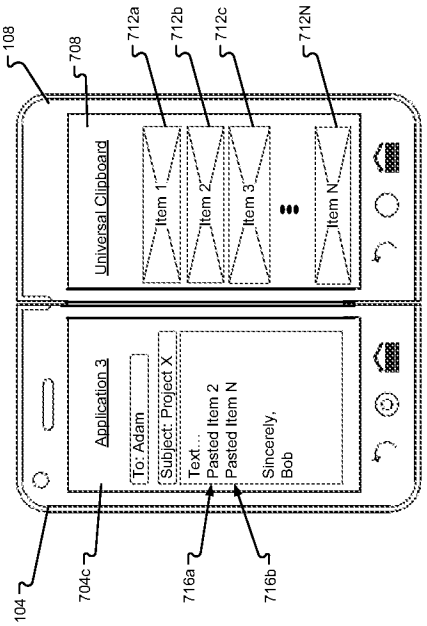


Fig. 7B

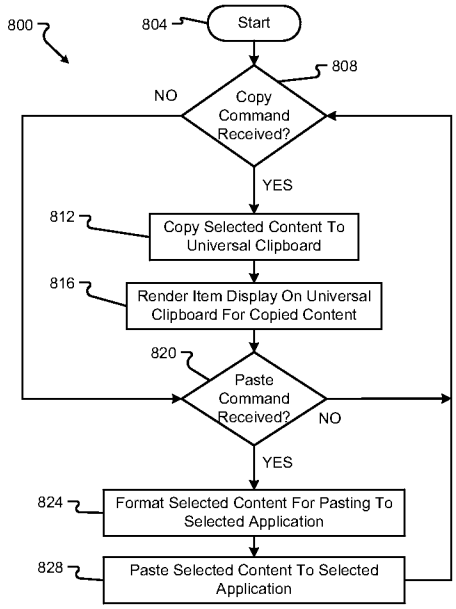


Fig. 8

SUMMARY

[0001] There is a need for a dual multi-display handheld computing device that provides for enhanced power and/or versatility compared to conventional single display handheld computing devices. These and other needs are addressed by the various aspects, embodiments, and/or configurations of the present disclosure. Also, while the disclosure is presented in terms of exemplary embodiments, it should be appreciated that individual aspects of the disclosure can be separately claimed.

[0002] Additionally, it is desirable to have the multi-display device be dockable or otherwise associatable with an additional device. In accordance with one exemplary embodiment, the multi-display device is dockable with a smartpad. The exemplary smartpad can include a screen, a touch sensitive display, a configurable area, a gesture capture region(s) and a camera. The smartpad can also includes a port adapted to receive the device. The exemplary smartpad is able to cooperate with the device such that information displayable on the device is also displayable on the smartpad. Furthermore, any one or more of the functions on the device are extendable to the smartpad, with the smartpad capable of acting as an input/output interface or extension of the smartpad. Therefore, for example, information from one or more of the displays on the multi-screen device is displayable on the smartpad.

[0003] Accordingly, an exemplary aspect is directed toward a smartpad comprising a screen, a display, a gesture capture area and at least one user configurable area, wherein the smartpad is dockable with a device, the device having two connected screens, wherein the smartpad is capable of displaying content from the two connected screens on the screen.

[0004] Another exemplary aspect is directed toward a method comprising docking a smartpad with a device, wherein the smartpad includes a screen, a display, a gesture capture area and at least one user configurable area, wherein the device has two connected screens; and displaying content from the two connected screens on the screen of the smartpad.

[0005] The present disclosure can provide a number of advantages depending on the particular aspect, embodiment, and/or configuration.

[0006] For example, the smartpad could provide additional display area to assist a user with viewing content on the device.

[0007] Additionally, power sharing and management functions are available due to, for example, the larger size of the smartpad that could be used for power storage.

[0008] Furthermore, the smartpad allows functions of a communications device to be extended to a tablet-like platform and/or form factor.

[0009] Even further, the disclosure provides techniques directed toward translating or otherwise converting content for a multi-display device into content for a single screen device.

[0010] Moreover, the smartpad provides additional input areas adapted to receive input beyond that of the touchscreen.

[0011] Additional advantages are directed toward managing the display of one or more windows associated with one or more applications on a multi-display device on a single display device.

[0012] Further advantages are directed toward emulation of multi-screen content on a single screen device.

[0013] These and other advantages will be apparent from the disclosure.

BRIEF DESCRIPTION OF THE DRAWINGS

[0014] FIG. 10 illustrates an exemplary smartpad (SP).

[0015] FIG. 11 illustrates an exemplary method of associating the smartpad with the device.

[0016] FIG. 12 illustrates a docked device with the smartpad.

[0017] FIGS. 13A – 13B illustrate an exemplary method for screen orientation.

[0018] FIG. 14 illustrates a method for displaying an application when the SP is in a landscape mode.

[0019] FIG. 15 illustrates a method for displaying an application when the SP is in a portrait mode.

[0020] FIG. 16 illustrates an example of a dual screen application in portrait max mode.

[0021] FIG. 17 illustrates an example of a dual screen application in max mode landscape.

[0022] FIG. 18 illustrates an example of keyboard management on the SP.

[0023] FIG. 19 illustrates an example of keyboard management on the SP with an application area in max mode.

[0024] FIG. 20 illustrates another example of keyboard management for the SP in landscape mode.

[0025] FIG. 21 illustrates an example of a dual screen application running in a dual screen emulation mode on the SP with a virtual keyboard.

[0026] FIG. 22 illustrates an example of application window stack management on the SP.

[0027] FIG. 23 illustrates another example of application window stack management on the SP.

[0028] FIG. 24 illustrates an example of multi application mode of the SP, wherein in the multi application mode the SP emulates the device in its mini-tablet form.

[0029] FIG. 25 illustrates another example of multi application mode of the SP.

[0030] FIG. 26 illustrates another example of multi application mode of the SP.

[0031] FIG. 27 illustrates another example of multi application mode of the SP.

[0032] FIG. 28 illustrates a method for managing screen display.

[0033] FIG. 29 illustrates an exemplary method for managing screen display with the desktop.

[0034] FIG. 30 illustrates an exemplary method of managing screen display with a keyboard.

[0035] FIGS. 31A and 31B illustrate desktop management on the SP.

[0036] FIGS. 32A and 32 B illustrate exemplary methods for desktop panel management.

[0037] FIG. 33 illustrates exemplary notification management on the SP.

[0038] FIGS. 34A and 34B illustrate exemplary techniques for application management.

[0039] FIGS. 35A and 35B illustrate an exemplary method for providing desktop previews or hints.

[0040] FIG. 36 illustrates an exemplary carousel application window stack.

[0041] FIG. 37 illustrates an exemplary carousel application window stack with a virtual keyboard.

[0042] FIG. 38 illustrates an exemplary method for associating the device and the SP.

[0043] FIG. 39 illustrates an exemplary method for application reorientation based on SP orientation.

[0044] FIG. 40 illustrates an exemplary method for managing the keyboard on the SP.

[0045] FIG. 41 illustrates an exemplary method for window manipulation based on one or more gestures.

[0046] FIG. 42 illustrates an exemplary method for application highlighting when an application is in focus in multi application mode.

[0047] FIG. 43 illustrates an exemplary method for application maximization.

[0048] FIG. 44 illustrates an exemplary method for transitioning from an application window to the desktop.

[0049] FIG. 45 illustrates an exemplary method for managing the display of the desktop and/or one or more panels on the SP.

[0050] FIG. 46 illustrates an exemplary method for merging panels for display on the SP.

[0051] FIG. 47 illustrates an exemplary method for previewing one or more panels on the SP.

[0052] FIG. 48 illustrates an exemplary method for stack management in multi application mode.

[0053] In the appended figures, similar components and/or features may have the same reference label. Further, various components of the same type may be distinguished by following the reference label by a letter that distinguishes among the similar components. If only the first reference label is used in the specification, the description is applicable to any one of the similar components having the same first reference label irrespective of the second reference label.

DETAILED DESCRIPTION

[0100] A multi-screen device 100 can receive activation of a window, in step 908. In embodiments, the multi-screen device 100 can receive activation of a window by receiving an input from the touch sensitive display 110 or 114, the configurable area 112 or 116, a gesture capture region 120 or 124, or some other hardware sensor operable to receive user interface inputs. The processor may execute the Task Management Module 540 may receive the input. The Task Management Module 540 can interpret the input as requesting an application task to be executed that will open a window in the window stack.

[0101] In embodiments, the Task Management Module 540 places the user interface interaction in the task stack 552 to be acted upon by the Display Configuration Module 568 of the Multi-Display Management Module 524. Further, the Task Management Module 540 waits for information from the Multi-Display Management Module 524 to send instructions to the Window Management Module 532 to create the window in the window stack.

[0102] The Multi-Display Management Module 524, upon receiving instruction from the Task Management Module 540, determines to which touch portion of the composite display 760, the newly activated window should be associated, in step 912. For example, window 4 770 is associated with the a portion of the composite display 764 In embodiments, the device state module 574 of the Multi-Display Management Module 524 may determine how the device is

oriented or in what state the device is in, e.g., open, closed, portrait, etc. Further, the preferences module 572 and/or requirements module 580 may determine how the window is to be displayed. The gesture module 576 may determine the user's intentions about how the window is to be opened based on the type of gesture and the location of where the gesture is made.

[0103] The Display Configuration Module 568 may use the input from these modules and evaluate the current window stack 760 to determine the best place and the best dimensions, based on a visibility algorithm, to open the window. Thus, the Display Configuration Module 568 determines the best place to put the window at the top of the window stack 760, in step 916. The visibility algorithm, in embodiments, determines for all portions of the composite display, which windows are at the top of the stack. For example, the visibility algorithm determines that window 3 768, window 4 770, and window 8 774 are at the top of the stack 760 as viewed in Figs. 7C through 7E. Upon determining where to open the window, the Display Configuration Module 568 can assign a display identifier 816 and possibly dimensions 808 to the window. The display identifier 816 and dimensions 808 can then be sent back to the Task Management Module 540. The Task Management Module 540 may then assign the window a stack position identifier 812 indicating the windows position at the top of the window stack.

[0104] In embodiments, the Task Management Module 540 sends the window stack information and instructions to render the window to the Window Management Module 532. The Window Management Module 532 and the Task Management Module 540 can create the logical data structure 800, in step 924. Both the Task Management Module 540 and the Window Management Module 532 may create and manage copies of the window stack. These copies of the window stack can be synchronized or kept similar through communications between the Window Management Module 532 and the Task Management Module 540. Thus, the Window Management Module 532 and the Task Management Module 540, based on the information determined by the Multi-Display Management Module 524, can assign dimensions 808, a stack position identifier 812 (e.g., window 1 782, window 4 770, etc.), a display identifier 816 (e.g., touch sensitive display 1 110, touch sensitive display 2 114, composite display identifier, etc,), and an active indicator 820, which is generally always set when the window is at the "top" of the stack. The logical data structure 800 may then be stored by both the Window Management Module 532 and the Task Management Module 540. Further, the Window Management Module

532 and the Task Management Module 540may thereinafter manage the window stack and the logical data structure(s) 800.

[0105] Demand for portable electronic devices with high levels of functionality continues to rise and personal electronic devices continue to become increasingly more portable. While computer power, battery life, screen size and overall functionality of portable phones and smart phones continues to increase, user reliance on these devices increases. Many users of such devices rely heavily on such devices for general communication, accessing the internet, cloud computing, and accessing various locally stored information such as contact information, files, music, pictures and the like. It is often desirable therefore to connect such heavily relied on devices to an additional computing device or display, such as a monitor or tablet device, such as a SmartPad (SP) 1000 (see Fig. 10).

[0106] Accordingly, it is desirable for the device 100 to be able to interface with an additional device, such as the SmartPad 1000, that enables functionality similar to, for example, both a tablet computer system and smart phone. Furthermore, a need exists for the above-described device to allow for various pre-existing features of both devices, such as sending and receiving phone calls and further allowing for the accessibility of applications running on the device 100. A need also exists for the above device 100 to provide the benefits of both a tablet computer system and cellular phone in one integrative device by allowing for common operations and functionality without compromising the form factor of the device.

[0107] One exemplary embodiment is directed toward a selectively removable device and smartpad system. The smartpad system is discussed in greater detail hereinafter, and can have various features for complementing the communications device, such as a smart phone or device 100. For example, the smartpad may supplement the device 100 by providing increased screen size, increased processor size, increased battery or power supply, or the like. Similarly, the device 100 may compliment the SP 1000 by providing connectivity through one or more wireless networks, access to various stored information, and the like. It will expressly recognized therefore that two or more devices of the present invention may be provided in a connected or docked and generally symbiotic relationship. It will further be recognized that the devices provide various features, benefits and functionality in their independent state(s).

[0108] In accordance with one exemplary embodiment, the device 100 is capable of being received by the SP 1000 through a recessed feature of the SP 1000 having corresponding

dimensions to the device 100. In one exemplary embodiment, the SP 1000 is provided and preferably sized for receiving a predetermined device 100. In alternative embodiments, however, it is contemplated that the SP 1000 is provided, the smartpad capable of receiving a plurality of communications devices of different sizes. In such embodiments, the SP 1000 may receive communications devices of various sizes by, for example, the inclusion of additional elements, such as spacers and various adjustable features.

[0109] In accordance with one exemplary embodiment, the device 100 and SP 1000 have a docking relationship that is established when the device 100 is connected to the SP 1000 during various modes of operation. For example, in one embodiment, a system is provided comprising the SP 1000 and the device 100, the SP 1000 capable of physically receiving the device 100, wherein the device 100 is operable as the primary computing device. In such an embodiment, the SP 1000 may, for example, simply provide enhanced audio and visual features for the device 100 that comprises its own CPU, memory, and the like. It is further contemplated that the system can be placed in a mode of operation wherein the device 100 docked to the SP 1000 provide it in a more passive mode where, for example, the device 100 draws power from the SP 1000 such as to recharge a battery of the device 100.

[0110] In accordance with another exemplary embodiment, the device 100 and SP 1000 are provided wherein the device 100 is received or docked with the SP 1000 and wherein a substantial area of the device 100 is positioned within one or more compartments of the SP 1000. For example, where as various known devices comprise docking features which require or result in the docked item to be generally exposed, thereby substantially altering the external dimensions of the host device and/or creating a potential for damaging one or both devices upon impact, an exemplary embodiment contemplates the SP 1000 which receives the device 100 in a manner such that the external dimensions of the SP 1000 are not substantially altered when the devices are connected. In such an arrangement, the device 100 and associated connection means are generally protected and the SP 1000 is allowed to substantially maintain its original shape. In accordance with one exemplary embodiment, the SP 1000 is capable of receiving and/or docking the device 100 wherein the device 100 is received in lockable association with the SP 1000. As used herein, the term "lockable" is not intended to designate or limit it to any particular arrangement. Rather, lockable is intended to refer to various embodiments as described herein and will be recognized by one of ordinary skill in the art. In one embodiment, the device 100 is

connectable to the SP 1000 wherein the SP 1000 comprises extension springs for first electively securing the device 100 in a docked manner and an ejection feature for releasing the device 100 ftom the SP 1000. Moreover, as will be described in greater detail below, it should be appreciated that the device 100 and SP 1000 can communicate using wired and/or wireless technology(ies) with equal success. Moreover, and in accordance with another exemplary embodiment, the hinged device 100 is selectively connectable to the SP 1000 wherein the device 100 is received by the SP 1000 in an open position and where in one or more preexisting ports of the SP 1000 correspond with internal receiving features of the SP 1000, such that the device 100 and the SP 1000 may be operated simultaneously in various modes of use.

[0111] In accordance with some exemplary embodiments, the SP 1000 is provided with an eject or release button to facilitate the removal of a stored or docked device 100.

[0112] Fig. 10 illustrates an exemplary SmartPad (SP) 1000 according to an exemplary embodiment. The exemplary SmartPad at least provides a larger touch sensitive display operatively coupleable to device 100.

[0113] While the following description uses the term "smart" in conjunction with the display device 1000, it is to be appreciated that this term does not necessarily connotate that there is intelligence in the SmartPad. Rather, it is to be appreciated that there can be "intelligence," including one or more of a processor(s), memory, storage, display drivers, etc., in the SmartPad, and/or one or more of these elements shared with the device 100 via, for example, one or more of a port, bus, connection, or the like. In general, any one or more of the functions of the device 100 is extendable to the SmartPad 700 and vice versa.

[0114] The exemplary SmartPad 700 includes a screen 1004, a SP touch sensitive display 1010, a SP configurable area 1008, a SP gesture capture region(s) 1012 and a SP camera 1016. The SP 1000 also includes a port (not visible in this orientation) adapted to receive the device 100 as illustrated at least in Fig. 11.

[0115] The device 100 docks with the SmartPad 1000 via the port on the SP 1000 and the corresponding port 136 on device 100. As discussed, port 136 in some embodiments is an input/output port (I/O port) that allows the device 100 to be connected to other peripheral devices, such as a display, keyboard, printing device and/or SP 1000. In accordance with one exemplary embodiment, the docking is accomplished by the device 100 sliding into the left-hand side of the SP 1000, with the device 100 being in an open state and the device 100 engaging a

port in the SP 1000 corresponding to port 136. In accordance with one exemplary embodiment, the device 100 engages a doored cassette-like slot in the SP 1000 into which the device 100 slides. (See for example Fig. 13) It should be appreciated however that there may be other configurations for physically and electrically engaging the two devices – in general, the manner of engagement is not important provided the device 100 and SP 1000 are in electrical communication with one another.

[0116] The SP 1000 includes a screen 1004. In some embodiments, the entire front surface of the SP 1000 may be touch sensitive and capable of receiving input by a user touching the front surface of the screen 1004. The screen 1004 includes touch sensitive display 1010, which, in addition to being touch sensitive, is also capable of displaying information to a user.

[0117] The screen 1004 also includes a configurable area 1008 that has been configured for specific inputs when the user touches portions of the configurable area 1008. Area 1012a is configured to receive a "back" input indicating that a user would like to view information previously displayed. Area 1012b is configured to receive a "menu" input indicating that the user would like to view options from a menu. Area 1012c is configured to receive a "home" input indicating that the user would like to view information associated with a "home" view.

[0118] In other embodiments, areas 1012a-c may be configured, in addition to the configurations described above, for other types of specific inputs including controlling features of device 100 and/or device 1000, some non-limiting examples including adjusting overall system power, adjusting the volume, adjusting the brightness, adjusting the vibration, selecting of displayed items on screen 1004, operating the SP camera 1016, operating a microphone, and initiating/terminating of telephone calls. Also, in some embodiments, areas 1012a-c may be configured for specific inputs depending upon the application running on device 100/ SP 1000 and/or information displayed on the touch sensitive displays 1010.

[0119] In addition to touch sensing, screen 1004 may also include areas that receive input from a user without requiring the user to touch the display area of the screen. For example, screen 1004 can include gesture capture area 1012. These areas are able to receive input by recognizing gestures made by a user without the need for the user to actually touch the surface of the display area. In comparison to touch sensitive display 1010 and 1014, the gesture capture area 1012 may not be capable of rendering a displayed image.

[0120] While not illustrated, there may also be a number of hardware components within SP 1000. As illustrated in Fig. 10, SP 1000 can include a speaker, a microphone and one or more cameras 1016. Upon docking the device 100 in the SP 1000, the corresponding device(s) (e.g., the speaker) in the device 100 could be disabled in favor of the speaker in the SP 1000. Similarly, other components, such as the screen 1004, microphone, speaker, etc, could be disabled on the device 100 in favor of the SP 1000.

[0121] In general, the touch sensitive display 1010 may comprise a full color, touch sensitive display. A second area within each touch sensitive screen 1004 may comprise the SP gesture capture region 1012. The SP gesture capture region 1012 may comprise an area or region that is outside of the SP touch sensitive display 1010 area that is capable of receiving input, for example in the form of gestures provided by a user. However, the SP gesture capture region 1012 does not necessarily include pixels that can perform a display function or capability.

[0122] A third region of the SP touch sensitive screen 1004 may comprise the configurable area 1012. The configurable area 1012 is capable of receiving input and has display or limited display capabilities. In embodiments, the configurable area 1012 may present different input options to the user. For example, the configurable area 1012 may display buttons or other relatable items. Moreover, the identity of displayed buttons, or whether any buttons are displayed at all within the configurable area 1012 of the SP touch sensitive screen 1004 may be determined from the context in which the device 1000 is used and/or operated. In an exemplary embodiment, the touch sensitive screen 1004 comprise liquid crystal display devices extending across at least those regions of the touch sensitive screen 1004 that is capable of providing visual output to a user, and a capacitive input matrix over those regions of the touch sensitive screen 1004 that is capable of receiving input from the user.

[0123] As discussed above with reference to Figs. 4A through 4H, the various graphical representations of gesture inputs that may be recognized by the screens 104, 108 are also recognizable by screen 1004. As discussed, the gestures may be performed not only by a user's body part, such as a digit, but also by other devices, such as a stylus, that may be sensed by the contact sensing portion(s) of a screen 1004. In general, gestures are interpreted differently, based on where the gestures are performed (either directly on the display 1004 or in the gesture capture region 1020). For example, gestures in the display 1010 may be directed to a desktop or application, and gestures in the gesture capture region 1020 may be interpreted as for the system.

[0124] In addition to the above, the SP touch sensitive screen 1004 may also have an area that assists a user with identifying which portion of the screen is in focus. This could be a bar of light or in general and indicator that identifies which one or more portions of the SP touch sensitive screen 1004 are in focus. (See for example, Fig. 29)

[0125] One or more display controllers (such as display controllers 216a, 216b and/or dedicated display controller(s) on the SP 1000) may be provided for controlling the operation of the touch sensitive screen 1004 including input (touch sensing) and output (display) functions.

[0126] In accordance with one exemplary embodiment, a separate touch screen controller is provided for the SP 1000 in addition to each of the controllers for the touch screens 104 and 108. In accordance with alternate embodiments, a common or shared touch screen controller may be used to control any one or more of the touch sensitive screens 104 and 108, and/or 1004. In accordance with still other embodiments, the functions of the touch screen controllers may be incorporated into other components, such as a processor and memory or dedicated graphics chip(s).

[0127] In a similar manner, the SP 1000 may include a processor complementary to the processor 204, either of which may comprise a general purpose programmable processor or controller for executing application programming or instructions. In accordance with at least some embodiments, the processors may include multiple processor cores, and/or implement multiple virtual processors. In accordance with still other embodiments, the processors may include multiple physical processors. As a particular example, the processors may comprise a specially configured application specific integrated circuit (ASIC) or other integrated circuit, a digital signal processor, a controller, a hardwired electronic or logic circuit, a programmable logic device or gate array, a special purpose computer, or the like. The processors generally function to run programming code or instructions implementing various functions of the device 100 and/or SP 1000.

[0128] The SP 1000 can also optionally be equipped with an audio input/output interface/device(s) (not shown) to provide analog audio to an interconnected speaker or other device, and to receive analog audio input from a connected microphone or other device. As an example, the audio input/output interface/device(s) 256 may comprise an associated amplifier and analog to digital converter usable with SP 1000. Alternatively or in addition, the device 100 can include an integrated audio input/output device 256 and/or an audio jack for interconnecting

an external speaker or microphone via SP 1000. For example, an integrated speaker and an integrated microphone can be provided, to support near talk or speaker phone operations.

[0129] Hardware buttons (not shown) but similar to hardware buttons 158 can be included for example for use in connection with certain control operations. Examples include a master power switch, volume control, etc., as described in conjunction with Figs. 1A through 1J. One or more image capture interfaces/devices 1016, such as a camera, can be included for capturing still and/or video images. Alternatively or in addition, an image capture interface/device 1016 can include a scanner or code reader. An image capture interface/device 1016 can include or be associated with additional elements, such as a flash or other light sources.

[0130] Communications between various components of the device 100 and SP 1000 can be carried by one or more buses and/or communications channels. In addition, power can be supplied to one or more of the components of the device 100 and Sp 1000 from a power source and/or power control module 260. The power control module 260 and/or device 100 and/or SP 1000 can, for example, include a battery, an AC to DC converter, power control logic, and/or ports for interconnecting the device 100/1000 to an external source of power.

[0131] The middleware 520 may also be any software or data that allows the multiple processes running on the devices to interact. In embodiments, at least portions of the middleware 520 and the discrete components described herein may be considered part of the OS 516 or an application 564. However, these portions will be described as part of the middleware 520, but those components are not so limited. The middleware 520 can include, but is not limited to, a Multi-Display Management (MDM) class 524, a Surface Cache class 528, a Window Management class 532, an Activity Management class 536, an Application Management class 540, a display control block, one or more frame buffers 548, an activity stack 552, and/or an event buffer 556 – all of the functionality thereof extendable to the SP 1000. A class can be any group of two or more modules that have related functionality or are associated in a software hierarchy.

[0132] The MDM class 524 also includes one or more modules that are operable to manage the display of applications or other data on the screen of the SP 1000. An embodiment of the MDM class 524 is described in conjunction with Fig. 5B. In embodiments, the MDM class 524 receives inputs from the OS 516, the drivers 512 and the applications 564. The inputs assist the MDM class 524 in determining how to display the information required by the user. Once a

determination for display configurations is determined, the MDM class 524 can bind the applications 564 to a display configuration. The configuration may then be provided to one or more other components to generate the display on the SP 1000.

[0133] Fig. 11 illustrates an exemplary embodiment showing the device 100 docking with the SP 1000. More specifically, the device 100 is being inserted into a slot (not shown) on the SP 1000. On completion of the inserting of device 100 into SP 1000 (See Fig. 12), device 100 communicates with the SP 1000 via bus or other wired or wireless electrical means 1204. The device 100 is also connected with, for example, the camera/video camera 1016, microphone (Mic), and power port 1208.

[0134] In conjunction with the docking of device 100 with SP 1000, one or more of the devices can begin power management. For example, one or more of the device 100 and SP 1000 can include power supplies, such as batteries, solar, or in general any electrical supply, any one or more of which being usable to supply one or more of the device 100 and SP 1000. Furthermore, through the use of, for example, an AC power adaptor connected to port 1208, the SP 1000 can supply power to device 100, such as to charge device 100. It will be appreciated that the power management functionality described herein can be distributed between one or more of the device 100 and SP 1000, with power being sharable between the two devices.

[0135] In addition to power management functions, upon the device 100 being docked with the SP 1000, the displays on device 100 can be turned off to, for example, save power. Furthermore, electrical connections are established between the device 100 and SP 1000 such that the speaker, microphone, display, input capture region(s), inputs, and the like, received by SP 1000 are transferrable to device 100. Moreover, the display on device 1000 is enabled such that information that would have been displayed on one or more of the touch sensitive displays 110 and 114 is displayed on touch sensitive display 1010. As will be discussed in greater detail herein, the SP 1000 can emulate the dual display configuration of the device 100 on the single display 1010.

[0136] The SP 1000 can optionally be equipped with the headphone jack 1212 and power button 1216. Moreover, any hardware buttons or user input buttons on the device 100 could be extended to and replicated on the SP 1000.

[0137] This dock event between the device 100 and SP 1000 can be seen as states 336 or 344 in Fig. 3A. As will be appreciated, and in accordance with one of the illustrative embodiments

herein, the device 100 is docked with SP 1000 with the device being in the open state 210. However, it is to be appreciated that the device 100 can be docked with the SP 1000 in the closed state 304, or docked via, for example, a cable without the device 100 necessarily being inserted into the SP 1000.

[0138] Figs. 13A-B illustrate application reorientation according to an exemplary embodiment of the invention. In particular, Fig. 13A illustrates the device 100 being inserted into the SP 1000. Before being associated with the SP 1000, the device 100 has two applications, both in the landscape mode, represented by application "B" in landscape on a first screen and application "C" in landscape on a second screen (partially obscured by SP 1000).

[0139] Fig. 13B illustrates the re-orientation of the windows for the two applications based on the device 100 being associated with the SP 1000, the SP 1000 being in the landscape orientation. In accordance with this exemplary embodiment, application "B" on the device 100 is re-oriented to be in the portrait orientation on the SP 1000, and in a similar manner, application "C" on the device 100 is reoriented to the portrait orientation on the right-hand side the touch sensitive display 1010. As will be appreciated, the reorientation of the application(s) from the device 100 to the SP 1000 can occur in a similar manner for a single application running on the device 100. For example, if there is only one application running on device 100, and the application is running in landscape mode, when the device 100 is docked with the SP 1000, the orientation of the application is reoriented to be appropriate for the current orientation of the SP 1000. For example, if the application on the device 100 is in portrait mode, and the SP 1000 is in landscape mode, the application is reoriented from portrait mode on the device 100 to landscape mode on the SP 1000. In a similar manner, if the application on the device is in landscape mode, and upon being docked to the SP 1000 in portrait mode, the application is reoriented into portrait mode for appropriate viewing on the SP 1000.

[0140] In accordance with one exemplary embodiment, the accelerometer 176 on device 100 is used to determine the orientation of both the device 100 and SP 1000, and consequently the orientation of the touch screen display 1010. Therefore, the accelerometer(s) 176 outputs a signal that is used in connection with the display of information to control the orientation and/or format in which information is to be displayed to the user on display 1010. As is to be appreciated, reorientation can include one or more of a portrait to landscape conversion, a

landscape to portrait conversion, a resizing, a re-proportioning and/or a redrawing of the window(s) associated with the application(s).

[0141] On reorienting of the running application(s), the application(s) is displayed on display 1010 on SP 1000.

[0142] In accordance with an optional exemplary embodiment, priority can be given to the application that is in focus. For example, and using again applications "B" and "C" as illustrated in Fig. 13B, if instead application C was in focus before docking, application C could be reoriented and displayed on the left-hand portion of display 1010, and application B, which was not in focus before docking, displayed on the right-hand portion of display 1010 upon docking.

[0143] In accordance with another optional embodiment, the application in focus could be displayed in full-screen mode on display 1010 with the application(s) not in focus placed into a window stack that is, for example, in a carousel-type arrangement as discussed hereinafter.

[0144] Fig. 14 illustrates an exemplary embodiment of a single application mode for the SP 1000. In the single application mode, all applications are launched and displayed in full screen. The single application mode can be indicated by a multi-tasking icon in the enunciator bar, or at some other location on screen 1004.

[0145] Displaying of the application(s) are managed by one or more of the display controller 544, framework 520, window management module 532, display configuration module 568, as well as middleware 520 and associated classes. In single application mode, all dual screen capable applications can be launched in either a dual screen or max mode, where the application is displayed substantially filling the display 1010. This is applicable to when the SP 1000 is either in the portrait mode, as illustrated in Fig. 14, or in the landscape mode, as illustrated in Fig. 15. In these figures, the "A" represents the single application with the X1, X2 being variables representing the coordinates and/or location of the window in which the application "A" is to be displaced. A similar notation is used hereinafter for the multi-application mode, with it being appreciated that, for example, X1 may contain the coordinate information for the displaying of the window for a first application, and X2 may contain the coordinate information for the displaying of a window corresponding to a second application, and so on.

[0146] Therefore, in one exemplar embodiment, when a single application is executed, a single application can launch in the full screen mode and can be correlated to the max mode as discussed in relation to Fig. 6I where a single application spans both screens of the device 100.

This max mode is applicable to both the portrait and landscape orientations as illustrated in Figs. 14 and Fig. 15 with the display configuration module 568 appropriately (re)sizing the window for the application to fit on substantially all or all of the display 1010.

[0147] This resizing can occur regardless of whether a native application on the device 100 actually supports the orientation of the SP 1000. Therefore, even if the application does not support a particular orientation on device 100, the display configuration module 568 can appropriately re-render and/or re-size the window for the application for appropriate display on the SP 1000.

[0148] Fig. 16 and Fig. 17 illustrate an exemplary method of rending a single application, that is a dual screen application, in the portrait max mode and landscape max mode, respectively. More specifically, in Fig. 16, the rendering of a dual screen application in portrait mode will display on display 1010 one of the two screens substantially or completely filling display 1010. A user then, for example using a gesture, could scroll between the two screens of the single application. In the landscape mode, as illustrated in Fig. 17, the screen 1010 is divided into a first portion 1704 and a second portion 1708. In this exemplary embodiment, the first screen of the dual screen application is rendered in first portion 1704, and the second screen of the dual screen application is rendered in the second portion 1708. While a certain portion of the screen 1010 is illustratively logically divided for the first portion 1704 and the second portion 1708, it should be appreciated that the screen real estate assigned to each portion can vary, for example, based on one or more of optimum display for the window(s), type of information being displayed in each portion, user preferences, rules associated with the application, and/or the like.

[0149] In accordance with a first example, the first portion is allocated one third of the screen 1010's resolution, while the second portion 1708 is allocated two thirds of the screen real estate. In accordance with another example, the screen 1010 is split 50/50. In accordance with yet another example, the first portion could be allocated 70% of the screen 1010's real estate, while the second portion 1708 could be allocated 30%. The managing and resizing of these windows can again be done in cooperation with the display configuration module 568, as well as the windows management module 532 and display controllers for successful rendering of the location of the window(s) on the SP 1000.

[0150] As will be appreciated, and in a manner similar to the operation of device 1000, should the SP 1000 change orientation (e.g., from landscape to portrait or vice versa) the window(s) for

the application(s) can be redrawn in the appropriate orientation taking into account window prioritization based on whether a particular application and current focus is for a dual screen application or a single screen application.

[0151] Focus can also be taken into consideration when determining which window of the application should be displayed when the SP 1000 is in the portrait position. For example, if the application is an e-mail client, and the application natively is displayed on dual screens on device 1000 (a first screen being directed toward showing inbox content, and the second screen being a preview window for a specific item in the inbox) the system can evaluate which window is currently in focus, and ensure that window is displayed in the portrait max mode when the SP 1000 is in the portrait orientation.

[0152] In Fig. 17 the SP 1000 is configured to merge windows ftom the dual screen application on to a single display 1010. In this landscape orientation, data (e.g., a single image, application, window, icon, video, etc.) from a first window is displayed in a first portion of the display 1010 while data (e.g., a single image, application, window, icon, video, etc.) is shown in a second portion of the display 1010. Similar to other output configurations, it may be possible to transition the SP 1000 from the shown output configuration to any other output configuration described herein, depending on, for example, into which state the SP 1000 is moved.

[0153] Some other exemplary embodiments of windows management within the SP 1000 upon the device 100 docking with the SP 1000 are as follows: For example, a device 100 is docked to the SP 1000, with the SP 1000 in a portrait orientation and there are two single-screen applications running on the device 1000, the application in focus is placed in a lower portion of the display 1010, and the application not in focus is placed on an upper portion of the display 1010. Another exemplary scenario, where the device 100 is docked to a portrait-oriented SP 1000 where one dual-screen application is running on the device 100 and the SP 1000 is in a dual application mode, applies gravity drop as discussed herein.

[0154] In another exemplary scenario, where the device 100 is running two single-screen applications, and the SP 1000 is in a landscape dual application mode, the first application is assigned to a first portion of the display 1010 and the second application is assigned to a second portion of the display 1010.

[0155] In yet another exemplary scenario where the device 100 is running one dual-screen application and the SP 1000 is in dual application landscape mode, both screens of the dual screen application can be shown on the SP 1000.

[0156] Stickiness can also apply to the SP 1000 such that, for example, when a first application is in focus, upon docking to a single application mode SP 1000, the application remains visible after docking. As another example of stickiness, if a second application is in focus upon docking to a single application mode SP 1000, application two remains visible after docking.

[0157] In accordance with another example, the device 100 is running one dual-screen application and is docked to a landscape-oriented SP 1000 in max mode, the windows are re-oriented to be side-by-side, opposed to one above the other.

[0158] Figs. 18 through Fig. 21 generally illustrate the management and display of a virtual keyboard 1804 on display 1010. More specifically, in Fig. 18, in portrait mode, the virtual keyboard 1804 is positioned below application area 1808, where an application is displayed in, for example, max mode. In general, it is preferred that the keyboard can be glued to the lower-portion of the display 1010, regardless of whether the SP is in the landscape or portrait mode. However, it is to be appreciated that, for example, based on user preferences, the screen can be glued to another portion of the screen, or can be moved to another location via, for example, a gesture. In Fig. 18, the application area 1808 displays, for example, a standard application with the virtual keyboard 1804 being displayed in the lower portion of display 1010. In Fig. 19, for example, the application area 1908 is showing a dual-screen enabled application in max mode. The keyboard 1804 is again similarly displayed in the lower portion of the display 1010.

[0159] In Fig. 20, in SP landscape mode, the keyboard 1804 is displayed in the lower portion of display 1010 with the application area 2004 substantially or completely filling the displayable area above the keyboard 1804. In Figure 21, the SP is again in landscape mode and displaying a dual-screen enabled application in max mode, the application area 1 2104 and application area 2 2108, the keyboard 1804 is displayed below the two application areas.

[0160] In general, in the embodiments illustrated in Fig. 18 through Fig. 21, a first determination is made as to whether a keyboard should be displayed. If the keyboard is to be displayed, the next determination is made as to the orientation of the SP. If the SP is in a portrait mode, the virtual keyboard is presented also in a portrait mode, preferable on the lower portion of the screen. If the SP is in a landscape mode, the keyboard is optionally re-sized to be

substantially displayed on a lower portion of the display with, for example, one or more application windows being located above the virtual keyboard. With the orientation of the SP change, the keyboard is also reoriented to be coincident with the orientation of the SP. Similarly, when the keyboard is no longer required, the keyboard is hidden with the application area(s) being expanded to again substantially fill the display 1010.

[0161] Fig. 22 and Fig. 23 illustrate exemplary methods of managing window positions on the SP 1000. In particular, in Fig. 22, application X 2204 is in view on display 1010. On receiving user input, such as the swipe motion represented by 2208 in the gesture capture region 1020, application X is "scrolled" to the left to be replaced with the dual-screen application A1|A2, as shown in Fig. 23. If the same gesture 2208 were to be repeated again, application Z would come into view. Similarly, if in Fig. 22 gesture 2208 was in the opposite direction, to the right, application Y would come into view on display 1010. Scrolling through available windows is of course applicable to both the landscape and portrait mode of the SP in a similar manner. For example, in portrait mode, instead of the gesture traversing ftom left to right or right to left, the gesture could traverse in a downward motion, or in an upward motion, with the virtual stacks of the windows being located "above" or "below" the device, similar to a rolodex. Thus, when the user initiates a downward type gesture, the next application "above" is displayed on display 1010.

[0162] Fig. 24 illustrates the multi application mode of the SP 1000, wherein in the multi application mode the SP 1000 emulates the device 100 in its mini-tablet form – with this mode optionally being invoked by selection of a multi application button (shown and described hereinafter). A simplified way of understanding this mode is to appreciate that the mode emulates the device 100 being opened. In this multi application mode, the SP 1000 can inherit the rules regarding the display of information on the device 100 - For example, that all applications are launched in single screen mode. One exception could be applications that support a max mode can be by default automatically expanded to this mode if provided the opportunity.

[0163] In this mode, each application has the ability to determine how the application appears in each orientation (e.g., portrait and landscape).

[0164] Fig. 26 illustrates an exemplary method of managing the multiple application mode of the SP 1000. In the multiple application mode, multiple applications can be managed and

displayed within the display 1010. In multi application mode, the SP1000 having the single screen emulates the dual screens of the device 100. To initiate the multiple application mode, a button/toggle 2618 is selected, which allows the user to select multiple applications for display in the display 1010. In this exemplary embodiment, a first application 2604 C, is shown in the upper-portion of the portrait mode SP 1000 and a second application 2608 D, is shown in a lower-portion of screen 1010. In conjunction with the displaying of multiple applications in the multiple application mode, focus indicator 2616 can be provided to assist the user with identifying which application is in focus. As discussed, this focus indicator can be a light bar, or other indicator (such as an indicator in the screen 1010 or beside 2608) drawing the user's attention to which application is in focus. In the exemplary embodiment in Fig. 26, application D 2608 is in focus as represented by the focus bar 2616. In accordance with this exemplary embodiment, and while the focus bar 2616 is shown in the gesture capture region 1020, it should be appreciated that the focus indicator could be located in some other portion of the SP 1000. For example, the window for the application in focus could be slightly re-sized to allow for the display of a bar of pixels adjacent to the window, which would similarly alert the user to the fact that that application is in focus. Similarly, the application in focus could appear at normal brightness while the application not in focus could be slightly dimmed. In general, any technique could be used to assist the user in readily determining which application is in focus.

[0165] To change focus, a user could use any of the gestures discussed herein or could, for example, simply touch the area where application C is displayed, thereby changing focus to application C, at which point a corresponding relocation of the focus indicator 2616 to adjacent to application C would occur.

[0166] Fig. 27 illustrates a similar scenario for a landscape mode SP 1000. In particular, and upon selection of the multi application mode, the display 1010 is divided between, in this example, a first application D 2712, and a second application F 2708. Here, application D is displayed on the right-hand portion of display 1010 and application F displayed on the left-hand portion of display 1010. While in this exemplary embodiment, the display real estate is split 50/50 between the two applications, it should be appreciated that one application could be displayed on a larger portion of the display 1010 than the other. In this particular exemplary embodiment, application D is in focus, as represented by focus indicator 2416.

[0167] In the multiple application mode, in both portrait and landscape orientations, each application could have its own associated window stack as show in Fig. 22 and Fig. 23, or there could be one stack shared between all of the displayed applications. More specifically, if each application has its own stack, with a stack structure similar to that illustrated in Fig. 22, a stack would be available for the first application, such as application C, and a similar stack would be available for application D. Each of these stacks could be independently scrolled through using, for example, a gesture as discussed above.

[0168] Fig. 28 illustrates an exemplary method for managing screen display characteristics according to another embodiment of this invention. In accordance with this embodiment, a determination is made whether an application can be maximized, and if it can be maximized, it is expanded to the dual screen mode or max mode, as appropriate, to substantially fill the display 1010 as illustrated in the figure. Here, application E1, which is an application that can be maximized, has been expanded using the max mode to substantially or completely fill display 1010.

[0169] In Fig. 28, button 2618 allows a user to toggle between a single screen mode (as illustrated in Fig. 28) and an emulated dual screen mode, for example, as illustrated in Fig. 26 and Fig. 27. Here, button 2618 does not include the "|" therefore indicating to the user the SP 1000 is in single screen mode.

[0170] Fig. 29 illustrates an exemplary method of managing windows. In this exemplary embodiment, and similar to the operation of the device 100, when the last application in the stack is moved to the side, the desktop is displayed. Even more specifically, as shown in Fig. 29, application F 2908 is displayed in an upper portion of display 1010 and the desktop 2912 is displayed in the lower portion of display 1010. Here the desktop is in focus, as illustrated by the focus indicator 2916. This configuration is available since the user has selected the dual-screen emulation mode button 2618.

[0171] Fig. 30 illustrates an exemplary method of displaying a keyboard according to one embodiment. In particular, when the SP is in portrait mode, the SP will have a keyboard area 3004 and an application area 3008. Upon display of the keyboard 3004, the application in application area 3008 is resized to substantially or completely fill the area of the screen not occupied by the keyboard 3004.

[0172] Fig. 31A and Fig. 31B illustrate desktop availability in both the single application mode and dual application mode in both the SP landscape mode and SP portrait mode. In particular, and in accordance with an exemplary embodiment, the desktop 3104 will occupy the entirety of the screen 1010. Additionally, and in accordance with this exemplary embodiment where the desktop is shown in a full-screen mode, the enunciator bar 1312 can be expanded across the entirety of the screen 1010. This can occur in both the portrait mode as shown in Fig. 31A as well as the landscape mode as illustrated in Fig. 31B. From here, upon selection of the application launcher 3116, the application launcher can optionally expand across the entirety of the screen 1010 in either the portrait or landscape mode. Similarly, the file explorer, which is launched by pressing the file explorer button 3120, can be similarly expanded into substantially all or all of the screen 1010 space.

[0173] Fig. 32A and Fig. 32B illustrate screen redrawing that may required to transition the desktop from the device 100 to the SP 1000. In particular, in Fig. 32A, six exemplary desktop panels are shown 3204-3224. These desktop panels are moveable in a carousel-like fashion based on gesture input from a user. However, it may not be possible to directly translate these panels to display correctly on the SP 1000 without the panels being distorted or not occupying the entirety of the screen 1010. Accordingly, in accordance with one exemplary embodiment, one or more of the panels 3204-3224 can be resized when displayed on the SP 1000 to accommodate all or substantially all of the screen 1010. In accordance with another exemplary embodiment, more than two of the panels can be shown on the screen 1010, such as a portion of panel D2 3208, a portion of panel D3 3212 and a portion of panel D4 3216. In this manner, the desktop illustrated on the SP 1000 will have a similar look-and-feel to the desktop panels shown on device 100. The same carousel-like motion is available via a gesture input to the SP 1000 such that a user can scroll to the one or more panels of the desktop.

[0174] Fig. 33 illustrates an exemplary notifications display that behaves on the SP 1000 in a very similar manner to that of the device 100. In particular, and whether in the portrait or landscape mode, notifications and status/toggle buttons are provided that allow the user to, for example, toggle on or off Bluetooth®, WiFi, screen lock, dual or single screen mode, power options, and the like. These, as well as other options, are generally shown in area 3304. Additional notifications can also be shown in area 3308 including one or more of carrier

information, notification information, communication information, and the like as discussed above.

[0175] In more detail, area 3304 provides some buttons for standard widgets such as WiFi toggle on and off, Bluetooth® toggle on and off, and the like. The screen lock toggle can allow, for example, user to lock the screen thereby prohibiting it from rotating despite the orientation of the SP 1000. The toggle can change color or display characteristics to indicate whether screen lock has been enabled. The single/dual application mode button toggles, for example, three different states including a dual application mode state, a single application mode state, and a single application lock state. The power mode toggle toggles between a handset optimized, SP optimized, hybrid power consumption, or of the like. These power modes can be associated with power levels, as indicated by the two battery status indicators 3312 and 3316 which correspond to the power levels in the device 100 and SP 1000, respectively.

[0176] Fig. 34A and Fig. 34B show exemplary methods of launching and displaying the application launcher upon selection of the application launcher button 3116. In both the portrait mode illustrated in Fig. 34A and landscape mode illustrated in 34B, and when the SP is in single application mode, the application launcher can appear in a bubble so as to not necessarily take up the entirety of the screen 1010. However, for example, upon receiving an appropriate gesture or input by the user, the application launcher can be expanded to take up the entirety of the screen 1010. This could further be classified in a preference such that the default is to always open the application launcher in a bubble so as to not take up the entirety of the screen 1010, or to always open in full screen mode, taking up substantially all or all of the screen 1010.

[0177] Fig. 35A and Fig. 35B illustrate an optional embodiment whereby the system could provide "hints" as to the content of desktop panels that are not fully in view. More specifically, Fig. 35A shows the first embodiment of the desktop panels D1-D6, 3504-3524 respectively. As discussed, a user is able to scroll through these panels in a carousel-type fashion, revealing any of the content in D1-D6. In accordance with the optional embodiment illustrated in Fig. 35B, where the same panels D1-D6 are available for viewing by user, the user is given a preview of one or more adjacent panels, here D2 3508 and D5 3520. In this configuration, this user is able to "peek" at the content of the adjacent panel(s) without necessarily scrolling in a carousel like fashion to view the entirety of the panel. More specifically, the entirety of panel D3 3512 and D4 3516 is shown on the display 1010. In addition, approximately one third of panel D2 3508

and one third of panel D5 3520 are also shown on the display 1010. As will be appreciated, more or less of panels D2 and D5 could be shown, with the greater of the panel being shown, the more visibility into that portion of the desktop the user would have.

[0178] Fig. 36 illustrates an optional embodiment for managing in a multiple application mode the stack of windows. This elegant method reinforces the conceptual model and assists the user with visualizing spatially the "carousel" type arrangement of the stack in the multi application mode. As is to be appreciated, while the exemplary embodiment shown in Fig. 36 is directed toward portrait mode of SP1000, it can work equally well in the landscape mode with the difference being instead of the carousel rotating from a top to bottom or bottom to top type arrangement, the "carousel" could be manipulated in a lefr to right or right to left type operation. However, the carousel could also work in a top to bottom or bottom to top mode in landscape mode as well.

[0179] In this exemplary embodiment, when multi-application mode is enabled via button 2618, application C 3604 and application D 3608 can be displayed, separated by separator 3612. In accordance with this optional exemplary embodiment, there are also one or more overflow applications behind application C 3604, here the overflow applications being 3424 and 3428. In a similar manner, there can be one or more overflow applications behind application D 3608, here application 3416 and application 3420. In this particular exemplary embodiment, the back and menu buttons 3432 can be enabled with a portion of the desktop 3436 being viewable behind the application stack. Upon receipt of one or more input gestures, such as gesture 3440, a user can scroll through the "carousel" of applications, in this instance, relocating application D 3608 to application C 3604's position, and thereby revealing application 3416. In addition, the focus indicator can be displayed near the application that is in focus. In this particular example, focus indicator 3444 is displayed beside application D. In accordance with an optional exemplary embodiment, instead of the stack "stopping" when the user reaches the last application, such as application 3420 or application 3428, the applications can be stacked in a circular manner, and continuously rotate in response to one or more input gestures by a user.

[0180] These concepts can be extended to the situation where the keyboard is also displayed in the multiple application mode. For example, and as illustrated in Fig. 37, the keyboard 3704 is displayed below the application stack 3708. "Behind" the application stack 3708, are additional applications 3712 and 3716 that the user can access via, for example, initiation of gesture 3720.

This exemplary embodiment has the applications scroll in a carousel-like fashion into and out of view, with the keyboard 3704 remaining visible, for example, at the bottom of display 1010. As with the embodiment in Fig. 35, the back and menu buttons can be enabled and, for example, virtually displayed on the display 1010 as illustrated by graphical button 3724.

[0181] Fig. 38 outlines an exemplary method for docking the device 100 with the smartpad 1000. In particular, control begins in step S3802 and continues to step S3804. In step S3804 the insertion of the device into the SP is detected. Next, in step S3806, power management can optionally begin. For example, and as discussed, the SP can be used as a power source for the device, the device can be used as a power source for the SP, power can be shared between the two devices and/or the SP can be plugged in, via for example, an AC adaptor, which is capable of charging one or more of the SP and the device 100. Then, in step S3808, the display of the device 100 is optionally turned off to, for example, conserve power. Control then continues to step S3810. In step S3810, communication and/or connectivity are established between the device 100 and the SP. Next, in step S3812, display and other input/output functions on the SP are enabled. Then, in step S3814, the device software settings are mapped to the smartpad hardware settings. Control then continues to step S3816.

[0182] In step S3816, the screen orientation of the device is automatically aligned to the orientation of the SP. Next, in step S3818, the last application in focus on the device remains in focus and is displayed on the SP. Normal operation and interaction with the SP then continues utilizing, for example, the same gestures as are usable with the device 100. Control then continues to step S3820 where the control sequence ends.

[0183] Fig. 39 illustrates an exemplary method of application/display orientation/reorientation. In particular, control begins in step S3900 and continues to step S3902. In step S3902, the orientation of the SP is detected. Next, in step S3904, the application orientation on the device is detected. Then, in step S3906, one or more displayed applications are reoriented to be in the same orientation as the SP. In addition, and based on the need for reorientation, a re-drawing or re-sizing of the application can also occur with the application(s) being displayed on the SP. Control then continues to step S3908 where the control sequence ends.

[0184] Fig. 40 outlines an exemplary method for managing the keyboard. In particular, control begins in step S4000 and continues to step S4002. In step S4002, determination is made as to whether a keyboard request has been detected. If a keyboard request has not been detected,

control jumps back to step S4000 with control otherwise continuing to step S4004. In step S4004, a determination is made as to the orientation of the SP. If the SP is in the landscape orientation, control jumps to step S4010 with control otherwise continuing to step S4006 with the SP being in the portrait orientation. In step S4006, the keyboard is displayed in the portrait mode.

[0185] Next, in step S4008, a determination is made as to whether there has been an orientation change. If there has been an orientation change, control jumps to step S4010 with control otherwise continuing to step S4014.

[0186] In step S4010, the keyboard is displayed in the landscape mode. Next, in step S4012, a determination is made as to whether there has been a change in orientation of the SP. If there has been a change in the orientation, control jumps to step S4006 with control otherwise continuing to S4014.

[0187] In step S4014, a determination is made as to whether the keyboard should be hidden. If the keyboard should be hidden, control continues to step S4016 with control otherwise continuing back to step S4004.

[0188] In step S4016, the keyboard is hidden with control continuing to step S4018 where the control sequence ends.

[0189] Fig. 41 illustrates an exemplary method for window management. In particular, control begins in step S4100 and continues to step S4102. In step S4102 a gesture is detected. As will be appreciated, and similar to the device 100, this gesture can be on the touch-sensitive display, in a configurable area, and/or in the gesture capture region(s). In step S4104, gesture direction can also optionally be detected. Then, in step S4106, the next application window can be brought into view, enabling, for example, a user to scroll through application window stack. A determination is then made as to whether another gesture has been detected. If another gesture has been detected, control jumps back to step S4104 with control otherwise continuing to step S4110.

[0190] Fig. 42 outlines an exemplary method for screen management. In particular, control begins in step S4200 and continues to step S4202. In step S4202, multiple application mode is enabled via, for example, selection of a button or toggle. Next, in step S4204, a user is optionally prompted to select which applications will be displayed. In an alternative embodiment, two adjacent applications in the stack are displayed, with one of the applications

being the application that is currently in focus. Then, in step S4206, the screen is split with the first portion of the screen displaying a first application and a second portion of the screen displaying a second application. Next, in S4208, the application that was detected to be in focus is then highlighted on the SP in step S4210. Control then continues to step S4212.

[0191] In step S4212, a determination is made as to whether a new application has been brought into focus. If a new application has been brought into focus, control jumps back to step S4210 where that application is highlighted with an "in-focus" indicator. Otherwise, control continues to step S4214 where the control sequence ends.

[0192] Fig. 43 outlines an exemplary method for windows management. In particular, control begins in step S4300 and continues to step S4302. In step S4302, a determination is made as to whether the application can be maximized. Next, in step S4304, if the application is maximizable, control jumps to step S4306 where the application is expanded to either the dual screen mode or the max mode. Control then continues to step S4308 where the control sequence ends.

[0193] Fig. 44 outlines an exemplary method for transitioning from an application window to the desktop. In particular, control begins in step S4400 and continues to step S4402. In step S4402, the last application in the application window stack is detected. Next, in step S4404, a gesture is detected, the gesture requesting a "next" window, however the current window is the last application in the window stack. In this scenario, in step S4406, the desktop is displayed in that there are no further windows to display in the application window stack. Control then continues to step S4408 where the control sequence ends.

[0194] Fig. 45 illustrates an exemplary method of emulating the multi-screen display of the device 100 on the SP1000. In particular, control beings in step S4500 and continues to step S4502. In step S4502, the desktop is displayed on the SP. Next, in step S4504, the desktop is logically divided on the SP into, for example, two sections. Then in step S4506, a first screen of the desktop is displayed in a first logical portion of the SP display. Then, in step S4508, a second screen of the desktop is displayed in a second logical portion of the SP display. Control then continues to step S4510.

[0195] In step S4510, carousel movement of the "panels" shown in the display can be initiated through user input, such as a gesture. Control then continues to step S4512 where the control sequence ends.

[0196] Fig. 46 outlines an exemplary method of displaying multiple "panels" of the desktop on the SP. In particular, control begins in step S4600 and continues to step S4602. In step S4602, a portion of the desktop is displayed on the SP. Next, in step S4604, the desktop is logically divided on the smartpad to accommodate multiple desktop "panels." Then, in step S4608, the first screen or panel of the desktop is displayed in one logical portion of the SP display. Then, in step S4610, a second screen or panel of the desktop is displayed bridging a first and a second logical portion of the SP display. Then, a third screen or panel of the desktop is displayed in the second logical portion of the SP display. Control then continues to step S4614.

[0197] In step S4614 carousel movement of the panels can be affected by, for example, an input of a gesture by the user. Control then continues to step S4616 where the control sequence ends.

[0198] Fig. 47 outlines an exemplary method of displaying one or more portions of the desktop. In particular, control begins in step S4700 and continues to step S4702. In step S4702, an access request to the desktop is detected. Next, in step S4704, at least one desktop panel is displayed. Then, in step S4706, at least one additional desktop panel is partially displayed on the desktop. Control then continues to step S4708.

[0199] In step S4708, and upon detection of a gesture, the partially displayed panel can be completely displayed on the display of the SP. Control then continues to step S4710 where the control sequence ends.

[0200] Fig. 48 outlines an exemplary method of windows management in multiple application mode. In particular, control begins in step S4800 and continues to step S4802. In step S4802, multiple application mode is entered. Next, in step S4804, the windows stack is arranged with one or more applications being partially visible behind a first application. Next, in step S4806, the stack can be arranged with one or more applications also partially being visible behind a second application. Then, in step S4808, and upon receiving an input gesture from a user, carousel-like scrolling can be enabled through the stack until the end of the stack is reached, or in a second embodiment, the stack can have a "circular" arrangement where continuous scrolling through the stack is possible. Control then continues to step S4810 where the control sequence ends.

[0201] The exemplary systems and methods of this disclosure have been described in relation to a smartpad and interaction with the device. However, to avoid unnecessarily obscuring the

present disclosure, the preceding description omits a number of known structures and devices. This omission is not to be construed as a limitation of the scopes of the claims. Specific details are set forth to provide an understanding of the present disclosure. It should however be appreciated that the present disclosure may be practiced in a variety of ways beyond the specific detail set forth herein.

[0202] For example, the smartpad could have multiple physical and/or logical screens/displays. Additionally, the smartpad could be used with one or more input devices such as a stylus, mouse, or the like. Moreover, the smartpad could be populated with a processor, memory, communications means and the like that would allow for stand-alone operation. Even further, the smartpad could be associated or docked with other types of communications devices such as a smartphone such that the smartpad could be used as a display and/or I/O interface therefore.

[0203] Furthermore, while the exemplary aspects, embodiments, and/or configurations illustrated herein show the various components of the system collocated, certain components of the system can be located remotely, at distant portions of a distributed network, such as a LAN and/or the Internet, or within a dedicated system. Thus, it should be appreciated, that the components of the system can be combined in to one or more devices, such as a tablet-like device, or collocated on a particular node of a distributed network, such as an analog and/or digital telecommunications network, a packet-switch network, or a circuit-switched network. It will be appreciated from the preceding description, and for reasons of computational efficiency, that the components of the system can be arranged at any location within a distributed network of components without affecting the operation of the system. For example, the various components can be located in a switch such as a PBX and media server, gateway, in one or more communications devices, at one or more users' premises, or some combination thereof. Similarly, one or more functional portions of the system could be distributed between a telecommunications device(s) and an associated computing device.

[0204] Furthermore, it should be appreciated that the various links connecting the elements can be wired or wireless links, or any combination thereof, or any other known or later developed element(s) that is capable of supplying and/or communicating data to and from the connected elements. These wired or wireless links can also be secure links and may be capable of communicating encrypted information. Transmission media used as links, for example, can be any suitable carrier for electrical signals, including coaxial cables, copper wire and fiber optics,

and may take the form of acoustic or light waves, such as those generated during radio-wave and infra-red data communications.

[0205] Also, while the flowcharts have been discussed and illustrated in relation to a particular sequence of events, it should be appreciated that changes, additions, and omissions to this sequence can occur without materially affecting the operation of the disclosed embodiments, configuration, and aspects.

[0206] In yet another embodiment, the systems and methods of this disclosure can be implemented in conjunction with a special purpose computer, a programmed microprocessor or microcontroller and peripheral integrated circuit element(s), an ASIC or other integrated circuit, a digital signal processor, a hard-wired electronic or logic circuit such as discrete element circuit, a programmable logic device or gate array such as PLD, PLA, FPGA, PAL, special purpose computer, any comparable means, or the like. In general, any device(s) or means capable of implementing the methodology illustrated herein can be used to implement the various aspects of this disclosure. Exemplary hardware that can be used for the disclosed embodiments, configurations and aspects includes computers, handheld devices, telephones (e.g., cellular, Internet enabled, digital, analog, hybrids, and others), and other hardware known in the art. Some of these devices include processors (e.g., a single or multiple microprocessors), memory, nonvolatile storage, input devices, and output devices. Furthermore, alternative software implementations including, but not limited to, distributed processing or component/object distributed processing, parallel processing, or virtual machine processing can also be constructed to implement the methods described herein.

[0207] In yet another embodiment, the disclosed methods may be readily implemented in conjunction with software using object or object-oriented software development environments that provide portable source code that can be used on a variety of computer or workstation platforms. Alternatively, the disclosed system may be implemented partially or fully in hardware using standard logic circuits or VLSI design. Whether software or hardware is used to implement the systems in accordance with this disclosure is dependent on the speed and/or efficiency requirements of the system, the particular function, and the particular software or hardware systems or microprocessor or microcomputer systems being utilized.

[0208] In yet another embodiment, the disclosed methods may be partially implemented in software that can be stored on a storage medium, executed on programmed general-purpose

computer with the cooperation of a controller and memory, a special purpose computer, a microprocessor, or the like. In these instances, the systems and methods of this disclosure can be implemented as program embedded on personal computer such as an applet, JAVA® or CGI script, as a resource residing on a server or computer workstation, as a routine embedded in a dedicated measurement system, system component, or the like. The system can also be implemented by physically incorporating the system and/or method into a software and/or hardware system.

[0209] Although the present disclosure describes components and functions implemented in the aspects, embodiments, and/or configurations with reference to particular standards and protocols, the aspects, embodiments, and/or configurations are not limited to such standards and protocols. Other similar standards and protocols not mentioned herein are in existence and are considered to be included in the present disclosure. Moreover, the standards and protocols mentioned herein and other similar standards and protocols not mentioned herein are periodically superseded by faster or more effective equivalents having essentially the same functions. Such replacement standards and protocols having the same functions are considered equivalents included in the present disclosure.

[0210] The present disclosure, in various aspects, embodiments, and/or configurations, includes components, methods, processes, systems and/or apparatus substantially as depicted and described herein, including various aspects, embodiments, configurations embodiments, subcombinations, and/or subsets thereof. Those of skill in the art will understand how to make and use the disclosed aspects, embodiments, and/or configurations afrer understanding the present disclosure. The present disclosure, in various aspects, embodiments, and/or configurations, includes providing devices and processes in the absence of items not depicted and/or described herein or in various aspects, embodiments, and/or configurations hereof, including in the absence of such items as may have been used in previous devices or processes, e.g., for improving performance, achieving ease and\or reducing cost of implementation.

[0211] The foregoing discussion has been presented for purposes of illustration and description. The foregoing is not intended to limit the disclosure to the form or forms disclosed herein. In the foregoing Detailed Description for example, various features of the disclosure are grouped together in one or more aspects, embodiments, and/or configurations for the purpose of streamlining the disclosure. The features of the aspects, embodiments, and/or configurations of

the disclosure may be combined in alternate aspects, embodiments, and/or configurations other than those discussed above. This method of disclosure is not to be interpreted as reflecting an intention that the claims require more features than are expressly recited in each claim. Rather, as the following claims reflect, inventive aspects lie in less than all features of a single foregoing disclosed aspect, embodiment, and/or configuration. Thus, the following claims are hereby incorporated into this Detailed Description, with each claim standing on its own as a separate preferred embodiment of the disclosure.

[0212] Moreover, though the description has included description of one or more aspects, embodiments, and/or configurations and certain variations and modifications, other variations, combinations, and modifications are within the scope of the disclosure, e.g., as may be within the skill and knowledge of those in the art, afrer understanding the present disclosure. It is intended to obtain rights which include alternative aspects, embodiments, and/or configurations to the extent permitted, including alternate, interchangeable and/or equivalent structures, functions, ranges or steps to those claimed, whether or not such alternate, interchangeable and/or equivalent structures, functions, ranges or steps are disclosed herein, and without intending to publicly dedicate any patentable subject matter.

What is claimed is:

1. A smartpad comprising:

a screen;

a display;

a gesture capture area; and

at least one user configurable area, wherein the smartpad is dockable with a device, the device having two connected screens, wherein the smartpad is capable of displaying content from the two connected screens on the screen.

2. The smartpad of claim 1, further comprising a docking interface, the docking interface adapted to receive the device and provide electrical communication between the smartpad and the device.

3. The smartpad of claim 1, wherein the display is a touch screen display.

4. The smartpad of claim 1, wherein the smartpad has a portrait mode and a landscape mode.

5. The smartpad of claim 1, wherein upon the device docking with the smartpad, one or more applications displayed on the device are oriented for display on the smartpad.

6. The smartpad of claim 1, wherein the smartpad splits the display into at least two logical portions.

7. The smartpad of claim 6, wherein a first of the logical portions displays a first application from the device and a second of the logical portions displays a second application from the device.

8. The smartpad of claim 1, wherein the smartpad has a single application mode, the single application mode displaying an application in full screen.

9. The smartpad of claim 8, wherein the single application mode displays the application in max mode.

10. The smartpad of claim 9, wherein in max mode, the application is oriented in a same orientation as the smartpad.

11. The smartpad of claim 1, further comprising one or more of a camera, a video camera, a microphone, a power button, a home button, a speaker, an AC adapter jack and a headphone jack.

12. A method comprising:

docking a smartpad with a device, wherein the smartpad includes a screen, a display, a gesture capture area and at least one user configurable area, wherein the device has two connected screens; and

displaying content from the two connected screens on the screen of the smartpad.

13. The method of claim 12, further comprising receiving the device in a docking interface of the smartad, the docking interface adapted to receive the device and provide electrical communication between the smartpad and the device.

14. The method of claim 12, wherein the display is a touch screen display.

15. The method of claim 12, wherein the smartpad has a portrait mode and a landscape mode.

16. The method of claim 12, wherein upon the device docking with the smartpad, one or more applications displayed on the device are oriented for display on the smartpad.

17. The method of claim 12, wherein the smartpad splits the display into at least two logical portions.

18. The method of claim 17, wherein a first of the logical portions displays a first application from the device and a second of the logical portions displays a second application from the device.

19. The method of claim 12, wherein the smartpad has a single application mode, the single application mode displaying an application in full screen.

20. One or more of one or more means for performing the steps of claim 12 and a non-transitory computer-readable information storage media having stored thereon instructions, that when executed by a processor, perform the steps of claim 12.

1. A smartpad comprising:

a screen; and

a display, the display configured to display content from a docked multi-screen device, wherein a dual screen application on the docked multi-screen device is displayed in full screen on the display.

2. The smartpad of claim 1, wherein the display is a touch screen display.

3. The smartpad of claim 1, wherein the smartpad has a portrait mode and a landscape mode.

4. The smartpad of claim 1, wherein the display is capable of being logically divided into at least two portions.

5. The smartpad of claim 6, wherein a first of the logical portions displays a first window from the multi-screen device and a second of the logical portions displays a second window from the multi-screen device.

6. The smartpad of claim 5, wherein the at least two portions are different sizes.

7. The smartpad of claim 1, wherein full screen occupies all or substantially all of the display.

8. A method comprising:

displaying content from a docked multi-screen device on a display of a smartpad, wherein a dual screen application on the docked multi-screen device is displayed in full screen on the display.

9. The method of claim 8, wherein the display is a touch screen display.

10. The method of claim 8, wherein the smartpad has a portrait mode and a landscape mode.

11. The method of claim 8, wherein the display is capable of being logically divided into at least two portions.

12. The method of claim 11, wherein a first of the logical portions displays a first window from the multi-screen device and a second of the logical portions displays a second window from the multi-screen device.

13. The method of claim 12, wherein the at least two portions are different sizes.

14. The method of claim 8, wherein full screen occupies all or substantially all of the display.

15. The method of claim 8, further comprising resizing the dual screen application.

16. The method of claim 8, further comprising redrawing the dual screen application.

17. The method of claim 8, further comprising determining an orientation of the smartpad and aligning an orientation of the dual screen application with an orientation of the smartpad.

18. The method of claim 8, further comprising adjusting proportions of the dual screen application.

19. One or more of one or more means for performing the steps of claim 8 and a non-transitory computer-readable information storage media having stored thereon instructions, that when executed by a processor, perform the steps of claim 8.

20. The method of claim 8, further comprising receiving a request to dock the multi-screen device on the smartpad, and, in response thereto, providing information from the multi-screen device on the display of the smartpad.

1. A smartpad comprising:
a screen; and
a display, the display configured to display content from a docked multi-screen device, the content including a virtual keyboard, wherein an application on the docked multi-screen device is displayed with the virtual keyboard on the display.

2. The smartpad of claim 1, wherein the display is a touch screen display.

3. The smartpad of claim 1, wherein the smartpad has a portrait mode and a landscape mode.

4. The smartpad of claim 1, wherein one or more of:
the display is capable of being logically divided into at least two portions,
the application is a dual screen application, and
the virtual keyboard has a landscape mode and a portrait mode.

5. The smartpad of claim 6, wherein a first of the logical portions displays a first window from the multi-screen device and a second of the logical portions displays a second window from the multi-screen device.

6. The smartpad of claim 5, wherein the at least two portions are different sizes and relate to two different applications.

7. The smartpad of claim 1, wherein the smartpad further supports a multi-application mode, the multi-application mode emulating the docked multi-screen device in mini-tablet form.

8. A method comprising:

displaying content from a docked multi-screen device on a display of a smartpad, the content including at least one virtual keyboard, wherein an application on the docked multi-screen device is displayed with the virtual keyboard on the display.

9. The method of claim 8, wherein the display is a touch screen display.

10. The method of claim 8, wherein the smartpad has a portrait mode and a landscape mode.

11. The method of claim 8, wherein one or more of:
the display is capable of being logically divided into at least two portions,
the application is a dual screen application, and
the virtual keyboard has a landscape mode and a portrait mode.

12. The method of claim 11, wherein a first of the logical portions displays a first window from the multi-screen device and a second of the logical portions displays a second window from the multi-screen device.

13. The method of claim 12, wherein the at least two portions are different sizes and relate to two different applications.

14. The method of claim 8, wherein the keyboard can have a portrait or landscape orientation.

15. The method of claim 8, further comprising resizing the application.

16. The method of claim 8, further comprising redrawing the application.

17. The method of claim 8, further comprising determining an orientation of the smartpad and aligning an orientation of the application with an orientation of the smartpad.

18. The method of claim 8, wherein the smartpad further supports a multi-application mode, the multi-application mode emulating the docked multi-screen device in mini-tablet form.

19. One or more of one or more means for performing the steps of claim 8 and a non-transitory computer-readable information storage media having stored thereon instructions, that when executed by a processor, perform the steps of claim 8.

20. The method of claim 8, further comprising receiving a gesture requesting viewing of a new application, and, in response thereto, scrolling though one or more available windows corresponding to available applications.

1. A smartpad comprising:

a screen; and

a display, the display configured to display content from a docked multi-screen device, the content including a first window corresponding to a first application and a second window corresponding to second application, wherein both windows are displayed on the display in a landscape orientation.

2. The smartpad of claim 1, wherein the display is a touch screen display.

3. The smartpad of claim 1, wherein the smartpad has a portrait mode and a landscape mode.

4. The smartpad of claim 1, wherein one or more of:

an application in focus is displayed with a focus indicator,

the display is capable of being logically divided into at least two portions,

the first application is a dual screen application, and

a virtual keyboard is displayed in a landscape mode or a portrait mode.

5. The smartpad of claim 4, wherein a first of the logical portions displays the first window from the multi-screen device and a second of the logical portions displays the second window from the multi-screen device.

6. The smartpad of claim 5, wherein the at least two portions are different sizes and relate to two different applications.

7. The smartpad of claim 1, wherein the smartpad further supports a multi-application mode, the multi-application mode emulating the docked multi-screen device in mini-tablet form.

8. A method comprising:

displaying content from a docked multi-screen device on a display of a smartpad, the content including a first window corresponding to a first application and a second window corresponding to second application, wherein both windows are displayed on the display in a landscape orientation.

9. The method of claim 8, wherein the display is a touch screen display.

10. The method of claim 8, wherein the smartpad has a portrait mode and a landscape mode.

11. The method of claim 8, wherein one or more of:

an application in focus is displayed with a focus indicator,

the display is capable of being logically divided into at least two portions,

the first application is a dual screen application, and

a virtual keyboard is displayed in landscape mode or a portrait mode.

12. The method of claim 11, wherein a first of the logical portions displays the first window from the multi-screen device and a second of the logical portions displays the second window from the multi-screen device.

13. The method of claim 12, wherein the at least two portions are different sizes and relate to two different applications.

14. The method of claim 8, wherein both windows occupy all or substantially all of the display.

15. The method of claim 8, further comprising resizing one or more of the windows.

16. The method of claim 8, further comprising redrawing one or more of the windows.

17. The method of claim 8, further comprising determining an orientation of the smartpad and aligning an orientation of one or more of the windows with the orientation of the smartpad.

18. The method of claim 8, wherein the smartpad further supports a multi-application mode, the multi-application mode emulating the docked multi-screen device in mini-tablet form.

19. One or more of one or more means for performing the steps of claim 8 and a non-transitory computer-readable information storage media having stored thereon instructions, that when executed by a processor, perform the steps of claim 8.

20. The method of claim 8, further comprising receiving a request to enter multi-application mode, and, in response thereto, splitting the display into at least two logical portions, a first portion for displaying the first window and a second portion for displaying the second window.

1. A smartpad comprising:
a screen; and
a display, the display configured to display content from a docked multi-screen device, the content including a first window corresponding to a first application and a second window corresponding to second application, wherein both windows are displayed on the display in a portrait orientation.

2. The smartpad of claim 1, wherein the display is a touch screen display.

3. The smartpad of claim 1, wherein the smartpad has a portrait mode and a landscape mode.

4. The smartpad of claim 1, wherein one or more of:
an application in focus is displayed with a focus indicator,
the display is capable of being logically divided into at least two portions,
the first application is a dual screen application, and
a virtual keyboard is displayed in a landscape mode or a portrait mode.

5. The smartpad of claim 4, wherein a first of the logical portions displays the first window from the multi-screen device and a second of the logical portions displays the second window from the multi-screen device.

6. The smartpad of claim 5, wherein the at least two portions are different sizes and relate to two different applications.

7. The smartpad of claim 1, wherein the smartpad further supports a multi-application mode, the multi-application mode emulating the docked multi-screen device in mini-tablet form.

8. A method comprising:

displaying content from a docked multi-screen device on a display of a smartpad, the content including a first window corresponding to a first application and a second window corresponding to second application, wherein both windows are displayed on the display in a portrait orientation .

9. The method of claim 8, wherein the display is a touch screen display.

10. The method of claim 8, wherein the smartpad has a portrait mode and a landscape mode.

11. The method of claim 8, wherein one or more of:
an application in focus is displayed with a focus indicator,
the display is capable of being logically divided into at least two portions,
the first application is a dual screen application, and
a virtual keyboard is displayed in landscape mode or a portrait mode.

12. The method of claim 11, wherein a first of the logical portions displays the first window from the multi-screen device and a second of the logical portions displays the second window from the multi-screen device.

13. The method of claim 12, wherein the at least two portions are different sizes and relate to two different applications.

14. The method of claim 8, wherein the smartpad is in a landscape orientation.

15. The method of claim 8, further comprising resizing one or more of the windows.

16. The method of claim 8, further comprising redrawing one or more of the windows.

17. The method of claim 8, further comprising determining an orientation of the smartpad and aligning an orientation of one or more of the windows with the orientation of the smartpad.

18. The method of claim 8, wherein the smartpad further supports a multi-application mode, the multi-application mode emulating the docked multi-screen device in mini-tablet form.

19. One or more of one or more means for performing the steps of claim 8 and a non-transitory computer-readable information storage media having stored thereon instructions, that when executed by a processor, perform the steps of claim 8.

20. The method of claim 8, further comprising receiving a request to enter multi-application mode, and, in response thereto, splitting the display into at least two logical portions, a first portion for displaying the first window and a second portion for displaying the second window.

1. A smartpad comprising:
a screen; and
a display, the display configured to display content from a docked multi-screen device, the content including a first window corresponding to a first application and a second window corresponding to second application, wherein both windows are displayed on the display and one window shows a desktop.

2. The smartpad of claim 1, wherein the display is a touch screen display.

3. The smartpad of claim 1, wherein the smartpad has a portrait mode and a landscape mode.

4. The smartpad of claim 1, wherein one or more of:
an application in focus is displayed with a focus indicator,
the display is capable of being logically divided into at least two portions,
the first application is a dual screen application, and
a virtual keyboard is displayed in a landscape mode or a portrait mode.

5. The smartpad of claim 6, wherein a first of the logical portions displays a first window from the multi-screen device and a second of the logical portions displays a second window from the multi-screen device.

6. The smartpad of claim 5, wherein the at least two portions are different sizes and relate to two different applications.

7. The smartpad of claim 1, wherein the smartpad further supports a multi-application mode, the multi-application mode emulating the docked multi-screen device in mini-tablet form.

8. A method comprising:

displaying content from a docked multi-screen device on a display of a smartpad, the content including a first window corresponding to a first application and a second window corresponding to second application, wherein both windows are displayed on the display and one window shows a desktop .

9. The method of claim 8, wherein the display is a touch screen display.

10. The method of claim 8, wherein the smartpad has a portrait mode and a landscape mode.

11. The method of claim 8, wherein one or more of:
an application in focus is displayed with a focus indicator,
the display is capable of being logically divided into at least two portions,
the first application is a dual screen application, and
a virtual keyboard is displayed in landscape mode or a portrait mode.

12. The method of claim 11, wherein a first of the logical portions displays a first window from the multi-screen device and a second of the logical portions displays a second window from the multi-screen device.

13. The method of claim 12, wherein the at least two portions are different sizes and relate to two different applications.

14. The method of claim 8, wherein the desktop includes selectable icons for one or more of applications, files and folders.

15. The method of claim 8, further comprising resizing one or more of the windows.

16. The method of claim 8, further comprising redrawing one or more of the windows.

17. The method of claim 8, further comprising determining an orientation of the smartpad and aligning an orientation of one or more of the windows with the orientation of the smartpad.

18. The method of claim 8, wherein the smartpad further supports a multi-application mode, the multi-application mode emulating the docked multi-screen device in mini-tablet form.

19. One or more of one or more means for performing the steps of claim 8 and a non-transitory computer-readable information storage media having stored thereon instructions, that when executed by a processor, perform the steps of claim 8.

20. The method of claim 8, further comprising receiving a request to enter multi-application mode, and, in response thereto, splitting the display into at least two logical portions, a first portion for displaying the first window and a second portion for displaying the second window.

1. A smartpad comprising:
a screen; and
a display, the display configured to display content from a docked multi-screen device, the content including at least a first desktop panel from a first screen of the docked multi-screen device and a second desktop panel from a second screen of the docked multi-screen device.

2. The smartpad of claim 1, wherein the display is a touch screen display.

3. The smartpad of claim 1, wherein the smartpad has a portrait mode and a landscape mode.

4. The smartpad of claim 1, wherein one or more of:
the display is capable of being logically divided into at least two logical portions,
one or more of one or more widgets and notifications are displayed, and
the panels are movable in a carousel-like fashion.

5. The smartpad of claim 4, wherein a first of the logical portions displays at least the first desktop panel and a second of the logical portions displays at least the second desktop panel.

6. The smartpad of claim 5, wherein the at least two portions are different sizes and relate to two different applications.

7. The smartpad of claim 1, wherein the smartpad further supports a multi-application mode, the multi-application mode emulating the docked multi-screen device in mini-tablet form.

8. A method comprising:

displaying content from a docked multi-screen device on a display of a smartpad, the content including at least a first desktop panel from a first screen of the docked multi-screen device and a second desktop panel from a second screen of the docked multi-screen device.

9. The method of claim 8, wherein the display is a touch screen display.

10. The method of claim 8, wherein the smartpad has a portrait mode and a landscape mode.

11. The method of claim 8, wherein one or more of:
the display is capable of being logically divided into at least two logical portions,
one or more of one or more widgets and notifications are displayed, and
the panels are movable in a carousel-like fashion.

12. The method of claim 11, wherein a first of the logical portions displays at least the first desktop panel and a second of the logical portions displays at least the second desktop panel.

13. The method of claim 12, wherein the at least two portions are different sizes and relate to two different applications.

14. The method of claim 8, wherein the desktop includes selectable icons for one or more of applications, files and folders.

15. The method of claim 8, further comprising resizing one or more of the panels.

16. The method of claim 8, further comprising redrawing one or more of the panels.

17. The method of claim 8, further comprising determining an orientation of the smartpad and aligning an orientation of one or more of the panels with an orientation of the smartpad.

18. The method of claim 8, wherein the smartpad further supports a multi-application mode, the multi-application mode emulating the docked multi-screen device in mini-tablet form.

19. One or more of one or more means for performing the steps of claim 8 and a non-transitory computer-readable information storage media having stored thereon instructions, that when executed by a processor, perform the steps of claim 8.

20. The method of claim 8, further comprising receiving a request to enter a desktop mode, and, in response thereto, splitting the display into at least two logical portions, displaying at least the first desktop panel from the first screen of the docked multi-screen device and displaying the second desktop panel from the second screen of the docked multi-screen device, and scrolling through the panels based on a received gesture.

1. A smartpad comprising:

a screen; and

a display, the display configured to display content from a docked multi-screen device, the content including at least a first desktop panel from a first screen of the docked multi-screen device and an application launcher.

2. The smartpad of claim 1, wherein the display is a touch screen display.

3. The smartpad of claim 1, wherein the smartpad has a portrait mode and a landscape mode.

4. The smartpad of claim 1, wherein one or more of:

the display is capable of being logically divided into at least two logical portions,

one or more of one or more widgets and notifications are displayed,

the panel and the application launcher are overlaid, and

the panel and application launcher are movable in a carousel-like fashion.

5. The smartpad of claim 4, wherein a first of the logical portions displays at least the first desktop panel and a second of the logical portions displays at least a second desktop panel.

6. The smartpad of claim 5, wherein the at least two portions are different sizes and relate to two different applications.

7. The smartpad of claim 1, wherein the smartpad further supports a multi-application mode, the multi-application mode emulating the docked multi-screen device in mini-tablet form.

8. A method comprising:

displaying content from a docked multi-screen device on a display of a smartpad, the content including at least a first desktop panel from a first screen of the docked multi-screen device and an application launcher.

9. The method of claim 8, wherein the display is a touch screen display.

10. The method of claim 8, wherein the smartpad has a portrait mode and a landscape mode.

11. The method of claim 8, wherein one or more of:

the display is capable of being logically divided into at least two logical portions,

one or more of one or more widgets and notifications are displayed,

the panel and the application launcher are overlaid, and

the panel and application launcher are movable in a carousel-like fashion.

12. The method of claim 11, wherein a first of the logical portions displays at least the first desktop panel and a second of the logical portions displays at least a second desktop panel.

13. The method of claim 12, wherein the at least two portions are different sizes and relate to two different applications.

14. The method of claim 8, wherein the desktop panel includes selectable icons for one or more of applications, files and folders.

15. The method of claim 8, further comprising resizing one or more of the panel and the application launcher.

16. The method of claim 8, further comprising redrawing one or more of the panel and the application launcher.

17. The method of claim 8, further comprising determining an orientation of the smartpad and aligning an orientation of the panel with an orientation of the smartpad.

18. The method of claim 8, wherein the smartpad further supports a multi-application mode, the multi-application mode emulating the docked multi-screen device in mini-tablet form.

19. One or more of one or more means for performing the steps of claim 8 and a non-transitory computer-readable information storage media having stored thereon instructions, that when executed by a processor, perform the steps of claim 8.

20. The method of claim 8, further comprising receiving a request to enter a desktop mode, and, in response thereto, splitting the display into at least two logical portions, and displaying at least the first desktop panel from the first screen of the docked multi-screen device and displaying the application launcher.

1. A smartpad comprising:
a screen; and
a display, the display configured to display content from a docked multi-screen device, the content including at least a first desktop panel from a first screen of the docked multi-screen device, a second desktop panel from a second screen of the docked multi-screen device and a portion of at least a third desktop panel.

2. The smartpad of claim 1, wherein the display is a touch screen display.

3. The smartpad of claim 1, wherein the smartpad has a portrait mode and a landscape mode.

4. The smartpad of claim 1, wherein one or more of:
the display is capable of being logically divided into at least two logical portions,
one or more of one or more widgets and notifications are displayed, and
the panels are movable in a carousel-like fashion.

5. The smartpad of claim 4, wherein a first of the logical portions displays at least the first desktop panel, a second of the logical portions displays at least the second desktop panel, and a third of the logical portions displays at least the third desktop panel.

6. The smartpad of claim 5, wherein the at least two portions are different sizes and relate to two different applications.

7. The smartpad of claim 1, wherein the smartpad further supports a multi-application mode, the multi-application mode emulating the docked multi-screen device in mini-tablet form.

8. A method comprising:
displaying content from a docked multi-screen device on a display of a smartpad, the content including at least a first desktop panel from a first screen of the docked multi-screen

device, a second desktop panel from a second screen of the docked multi-screen device and a portion of at least a third desktop panel.

9. The method of claim 8, wherein the display is a touch screen display.

10. The method of claim 8, wherein the smartpad has a portrait mode and a landscape mode.

11. The method of claim 8, wherein one or more of:
the display is capable of being logically divided into at least two logical portions,
one or more of one or more widgets and notifications are displayed, and
the panels are movable in a carousel-like fashion.

12. The method of claim 11, wherein a first of the logical portions displays at least the first desktop panel, a second of the logical portions displays at least the second desktop panel, and a third of the logical portions displays at least the third desktop panel.

13. The method of claim 12, wherein the at least two portions are different sizes and relate to two different applications.

14. The method of claim 8, wherein at least one desktop panel includes selectable icons for one or more of applications, files and folders.

15. The method of claim 8, further comprising resizing one or more of the panels.

16. The method of claim 8, further comprising redrawing one or more of the panels.

17. The method of claim 8, further comprising determining an orientation of the smartpad and aligning an orientation of the at least one of the panels with an orientation of the samrtpad.

18. The method of claim 8, wherein the smartpad further supports a multi-application mode, the multi-application mode emulating the docked multi-screen device in mini-tablet form.

19. One or more of one or more means for performing the steps of claim 8 and a non-transitory computer-readable information storage media having stored thereon instructions, that when executed by a processor, perform the steps of claim 8.

20. The method of claim 8, further comprising receiving a request to enter a desktop mode, and, in response thereto, splitting the display into at least two logical portions, displaying at least the first desktop panel from the first screen of the docked multi-screen device and displaying the second desktop panel from the second screen of the docked multi-screen device, wherein a preview of the portion of at least a third desktop panel is also displayed.

1. A smartpad comprising:
a screen; and
a display, the display configured to display content from a docked multi-screen device, the content including at least a first application window and a second application window, the smartpad running in a multi-application mode wherein at least a portion of a card stack representing at least one additional window is also displayed.

2. The smartpad of claim 1, wherein the display is a touch screen display.

3. The smartpad of claim 1, wherein the smartpad has a portrait mode and a landscape mode.

4. The smartpad of claim 1, wherein one or more of:
the display is capable of being logically divided into at least two logical portions,
a portion of a desktop is visible, and
the panels are movable in a carousel-like fashion based on a received gesture.

5. The smartpad of claim 4, wherein a first of the logical portions displays at least the first application window, a second of the logical portions displays at least the second application window, and one or more of the first logical portion and second logical portion also display the portion of the card stack.

6. The smartpad of claim 5, wherein the at least two portions are different sizes and relate to two different applications.

7. The smartpad of claim 1, wherein the smartpad further supports a multi-application mode, the multi-application mode emulating the docked multi-screen device in mini-tablet form.

8. A method comprising:

displaying content from a docked multi-screen device on a display of a smartpad, the content including at least a first application window and a second application window, the smartpad running in a multi-application mode wherein at least a portion of a card stack representing at least one additional window is also displayed.

9. The method of claim 8, wherein the display is a touch screen display.

10. The method of claim 8, wherein the smartpad has a portrait mode and a landscape mode.

11. The method of claim 8, wherein one or more of:
the display is capable of being logically divided into at least two logical portions,
a portion of a desktop is visible, and
the panels are movable in a carousel-like fashion based on a received gesture.

12. The method of claim 11, wherein a first of the logical portions displays at least the first application window, a second of the logical portions displays at least the second application window, and one or more of the first logical portion and second logical portion also display the portion of the card stack.

13. The method of claim 12, wherein the at least two portions are different sizes and relate to two different applications.

14. The method of claim 8, wherein a desktop includes selectable icons for one or more of applications, files and folders.

15. The method of claim 8, further comprising resizing one or more of the windows.

16. The method of claim 8, further comprising redrawing one or more of the windows.

17. The method of claim 8, further comprising determining an orientation of the smartpad and aligning an orientation of the at least one of the windows with an orientation of the samrtpad.

18. The method of claim 8, wherein the smartpad further supports a multi-application mode, the multi-application mode emulating the docked multi-screen device in mini-tablet form, the windows in the card stack appearing partially behind one or more of the first application window and the second application window.

19. One or more of one or more means for performing the steps of claim 8 and a non-transitory computer-readable information storage media having stored thereon instructions, that when executed by a processor, perform the steps of claim 8.

20. The method of claim 8, further comprising receiving a request to view another window in the card stack, the request being a gesture, and, in response thereto:
moving the first application window into a partially obscured position in the card stack and displaying the second application window,
moving the first application window into a partially obscured position in the card stack and displaying the at least one additional window, or
moving the first application window into a partially obscured position in the card stack and displaying a desktop.

ABSTRACT

A multi-display device is adapted to be dockable or otherwise associatable with an additional device. In accordance with one exemplary embodiment, the multi-display device is dockable with a smartpad. The exemplary smartpad can include a screen, a touch sensitive display, a configurable area, a gesture capture region(s) and a camera. The smartpad can also include a port adapted to receive the device. The exemplary smartpad is able to cooperate with the device such that information displayable on the device is also displayable on the smartpad. Furthermore, any one or more of the functions on the device are extendable to the smartpad, with the smartpad capable of acting as an input/output interface or extension of the smartpad. Therefore, for example, information from one or more of the displays on the multi-screen device is displayable on the smartpad.

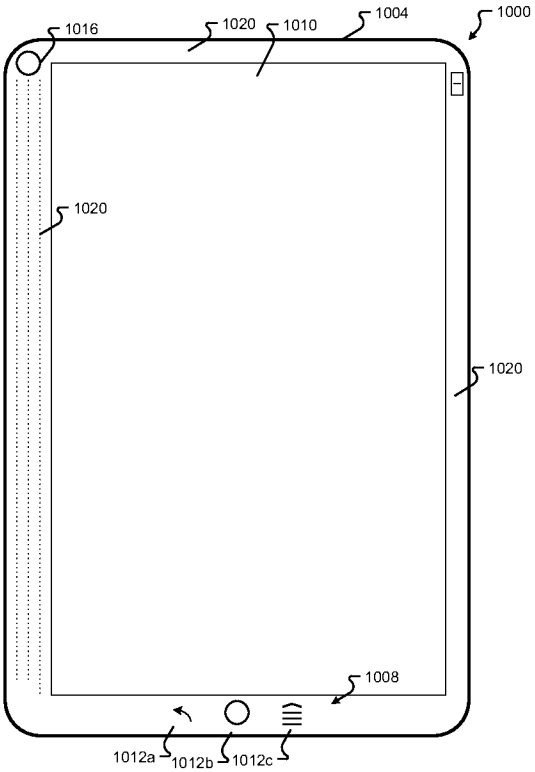


FIG. 10

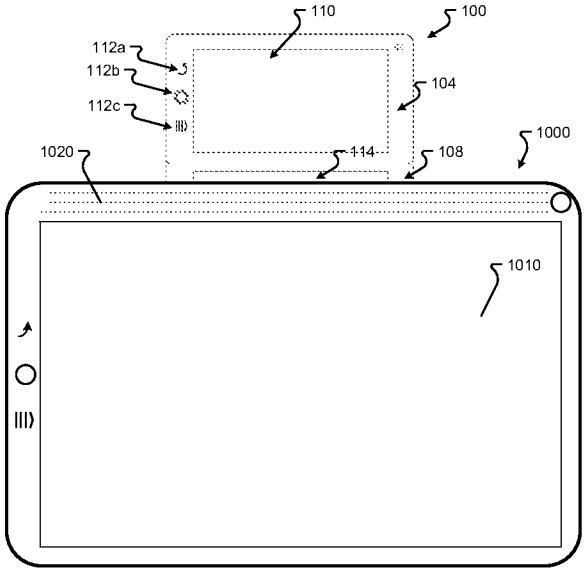


FIG. 11

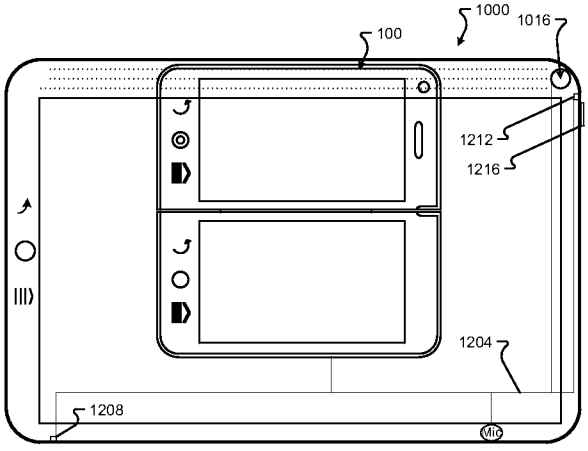


FIG. 12

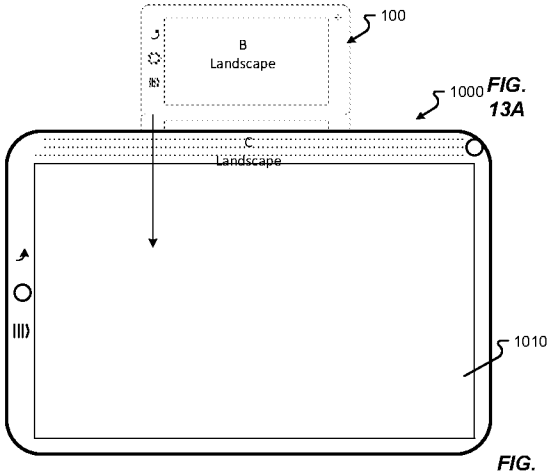


FIG. 13A

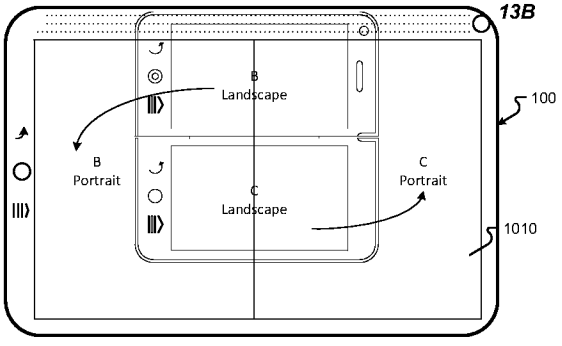


FIG. 13B

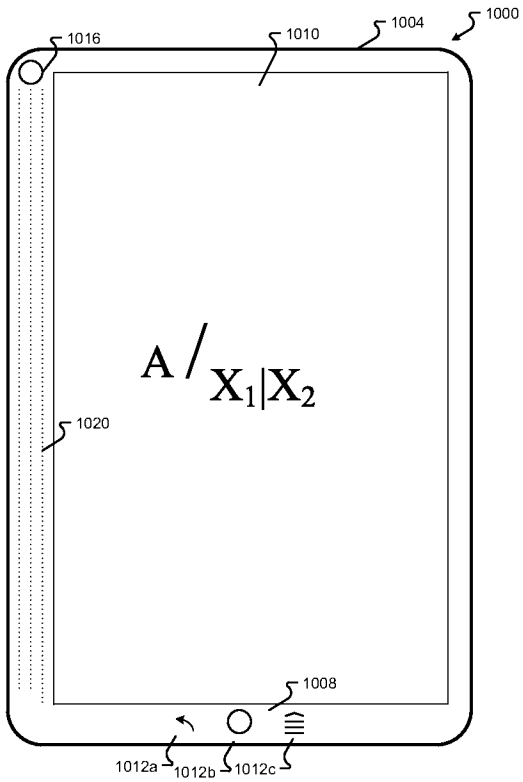


FIG. 14

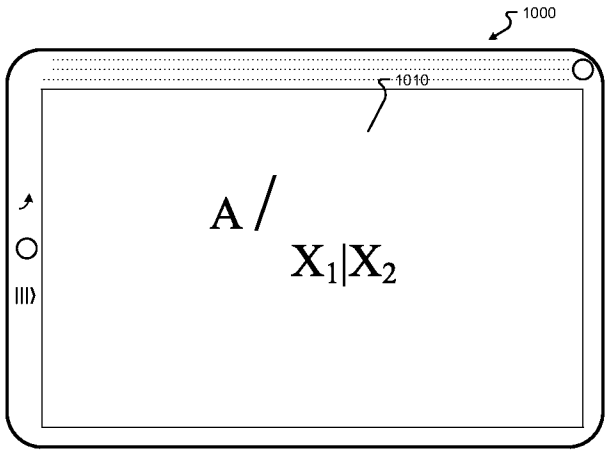


FIG. 15

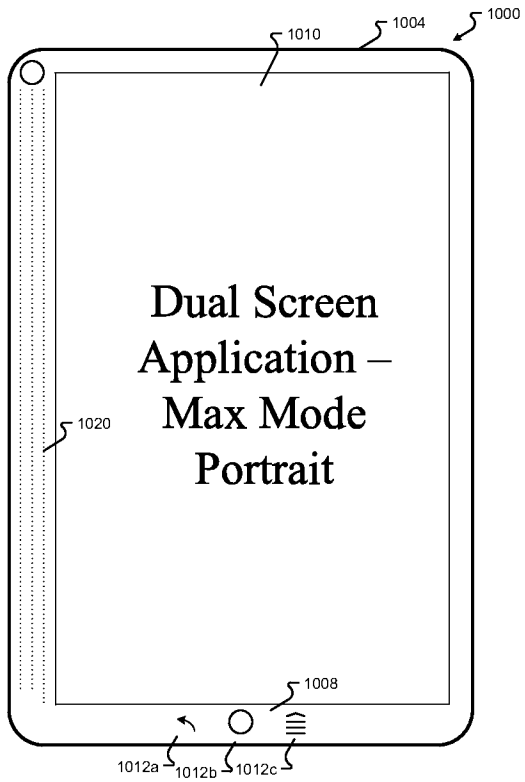


FIG. 16

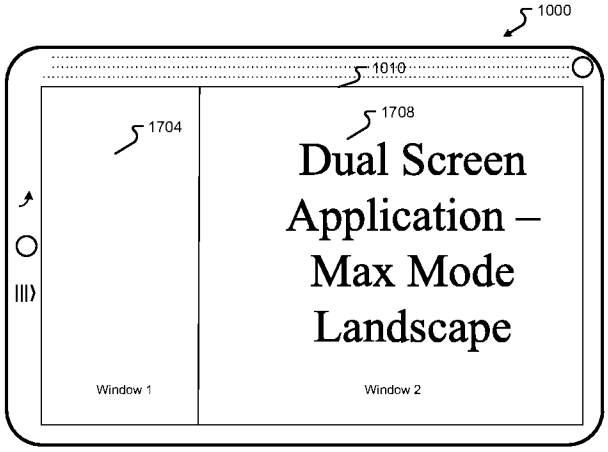


FIG. 17

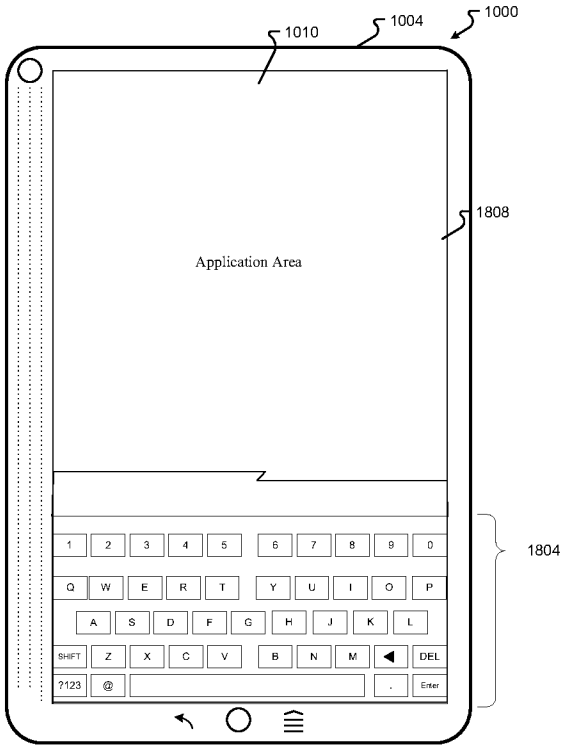


FIG. 18

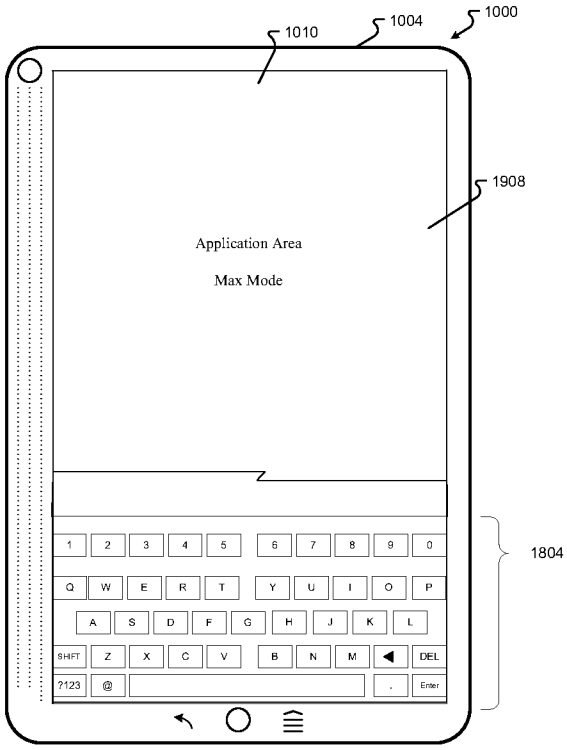


FIG. 19

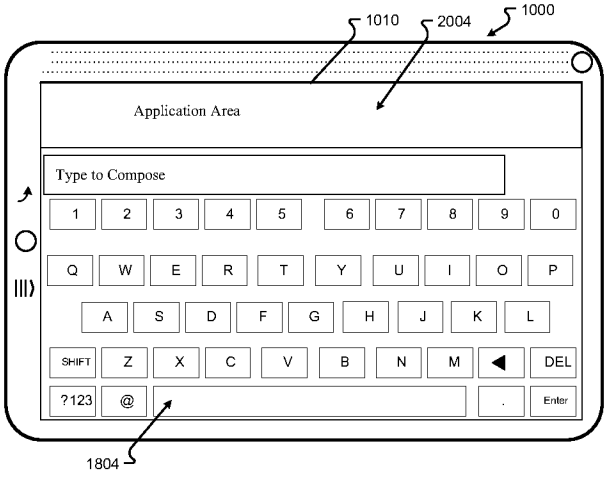


FIG. 20

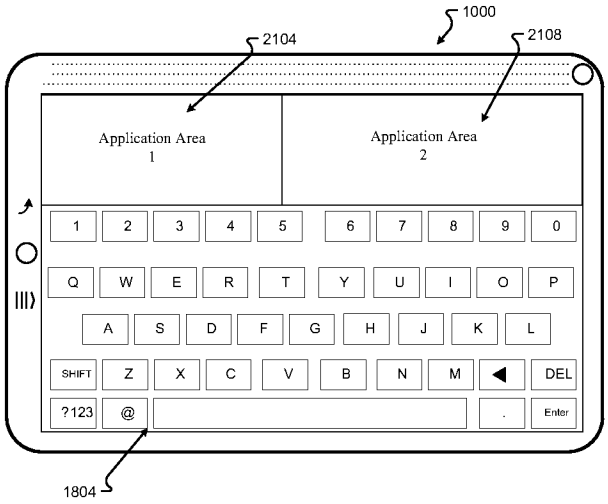


FIG. 21

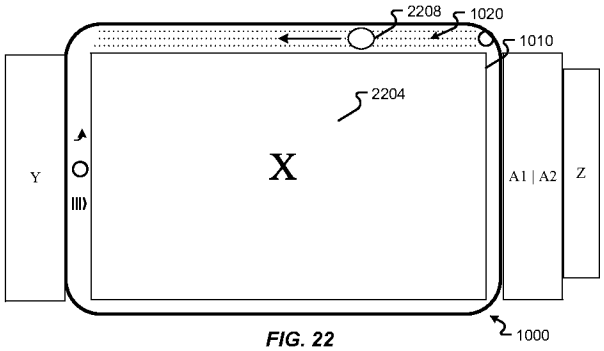


FIG. 22

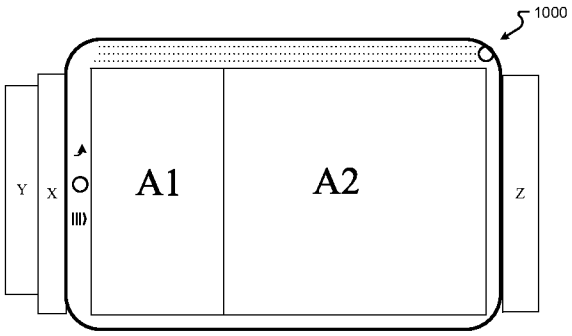


FIG. 23

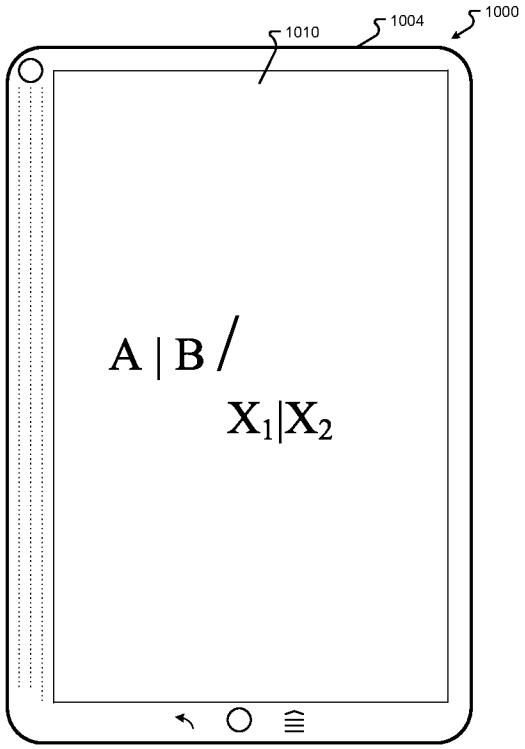


FIG. 24

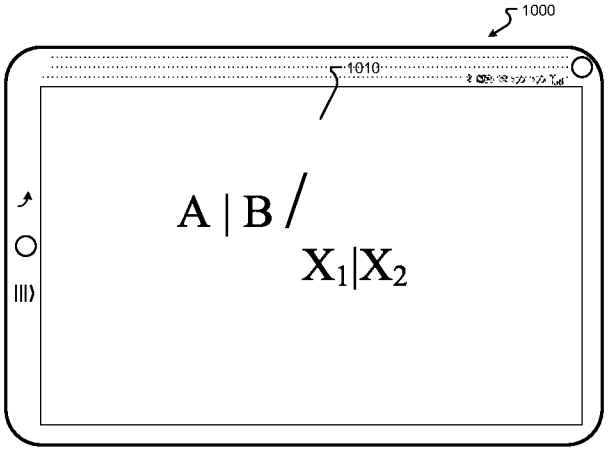


FIG. 25

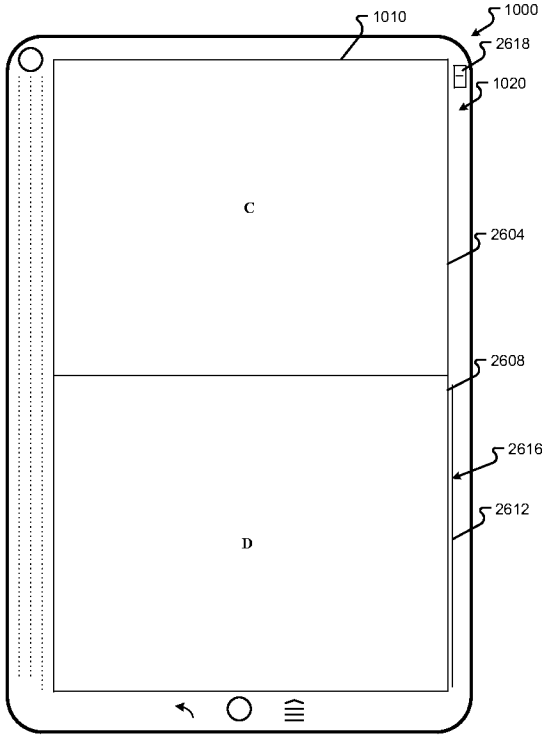


FIG. 26

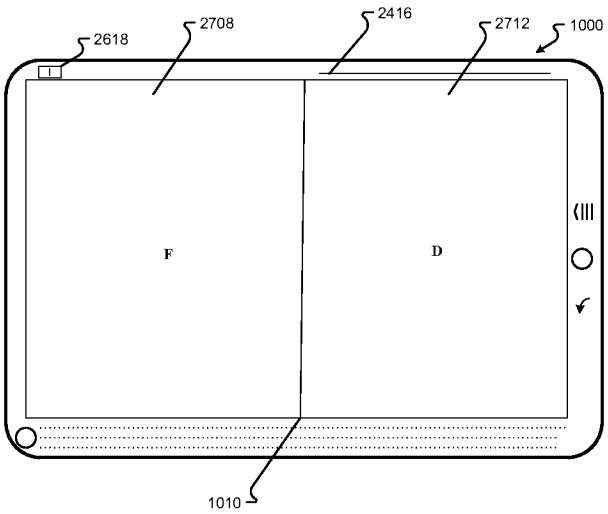


FIG. 27

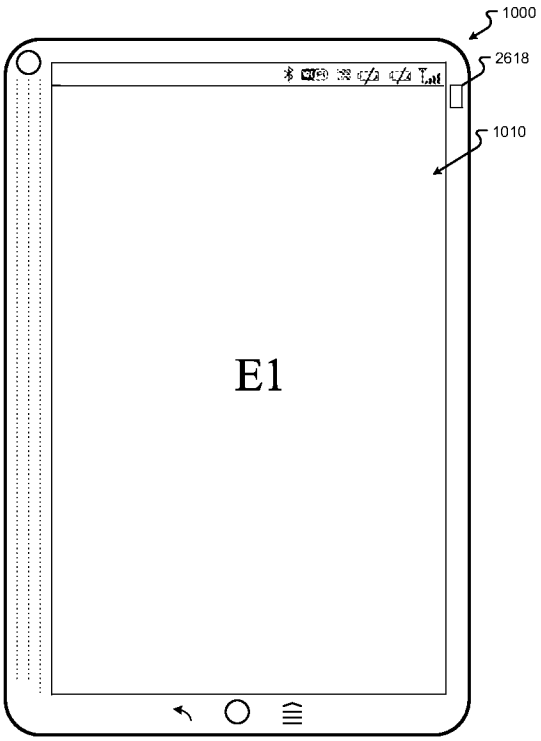


FIG. 28

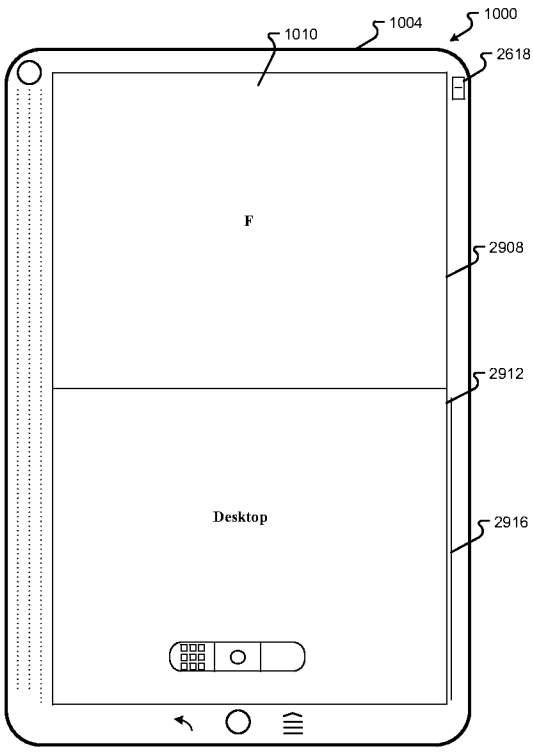


FIG. 29

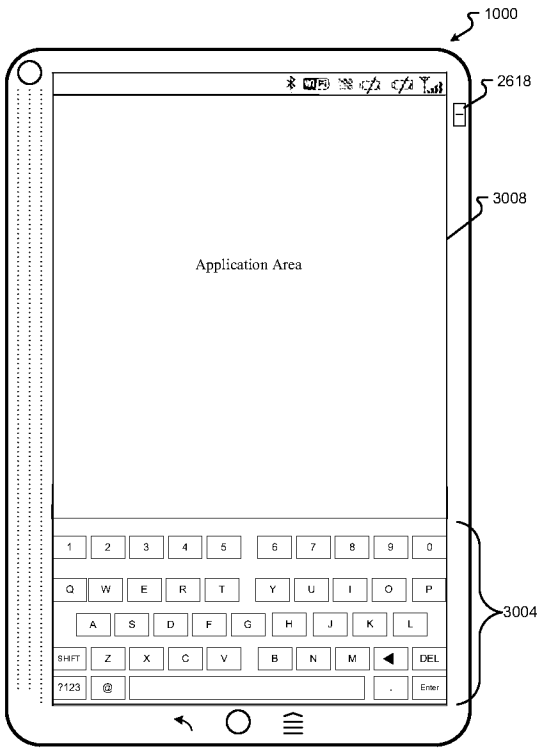


FIG. 30

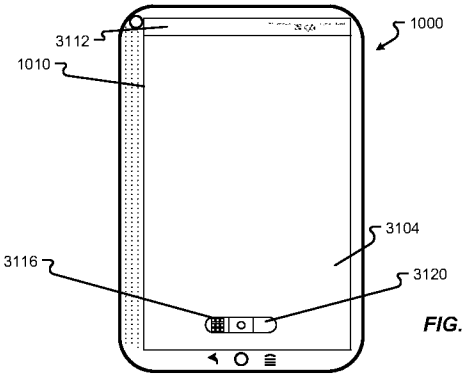


FIG. 31A

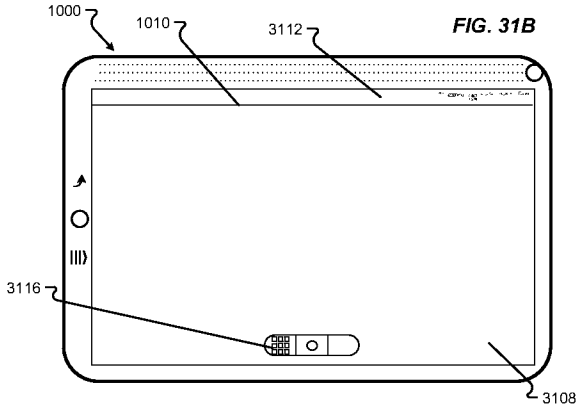


FIG. 31B

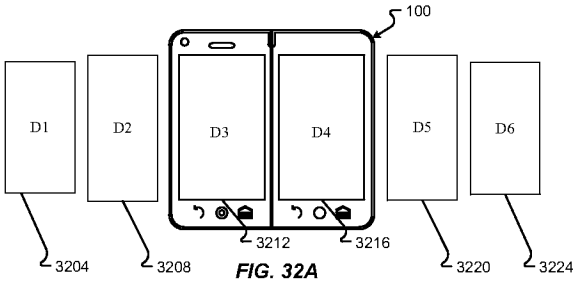


FIG. 32A

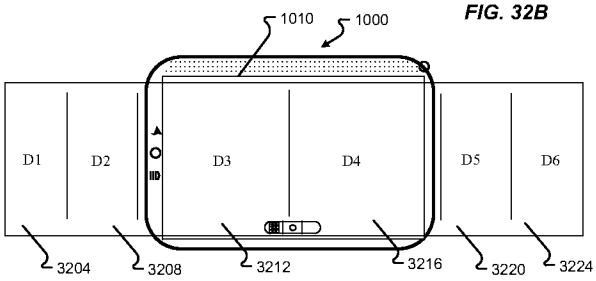


FIG. 32B

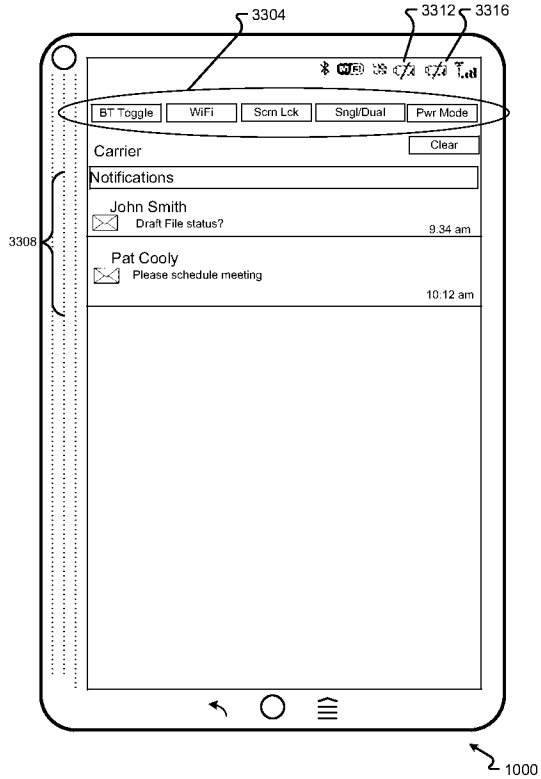


FIG. 33

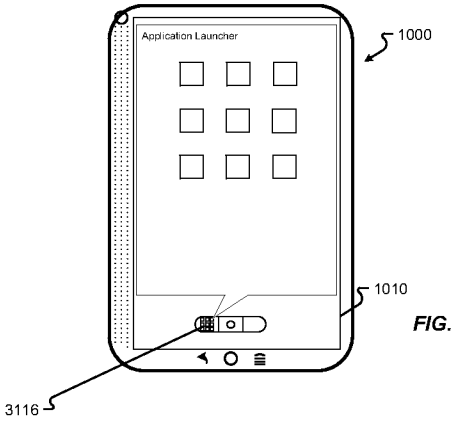


FIG. 34A

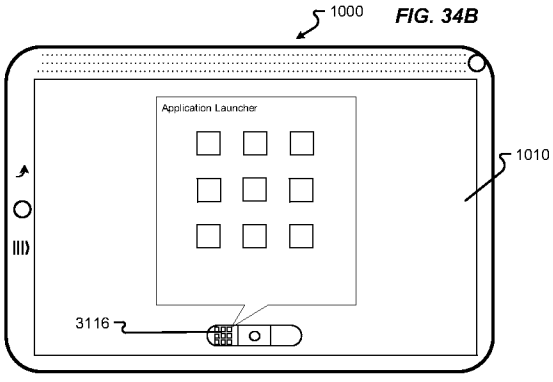


FIG. 34B

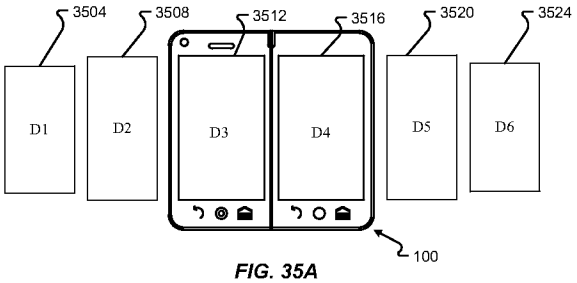


FIG. 35A

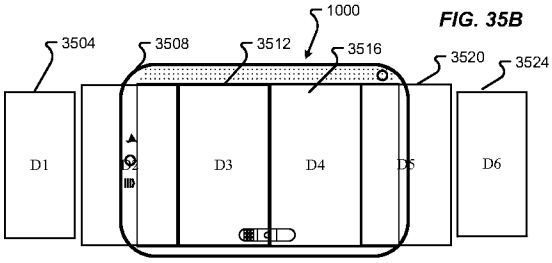


FIG. 35B

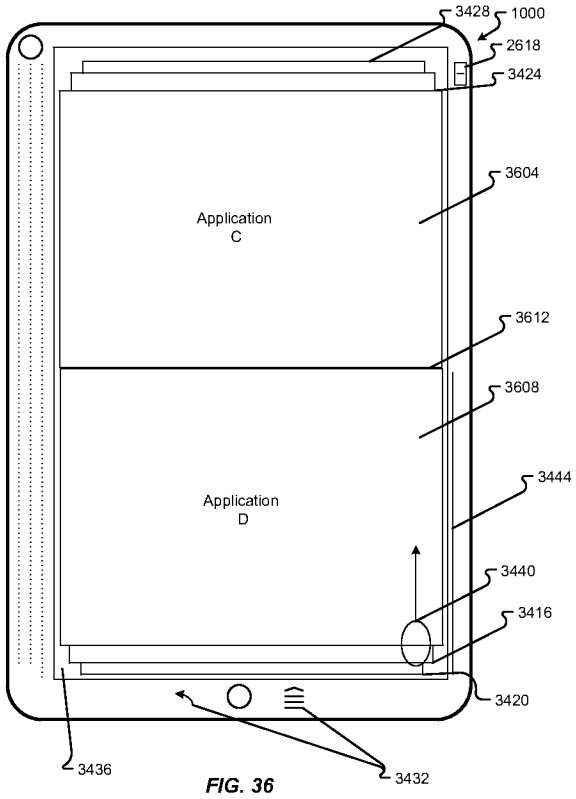


FIG. 36

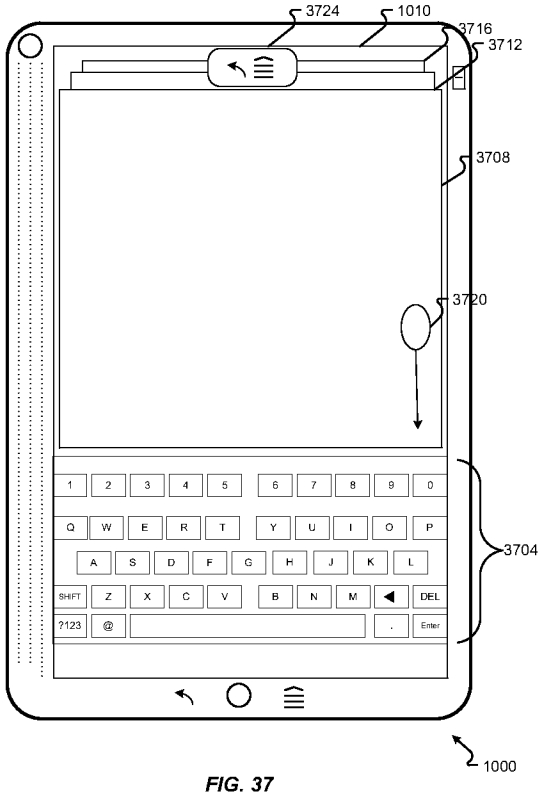


FIG. 37

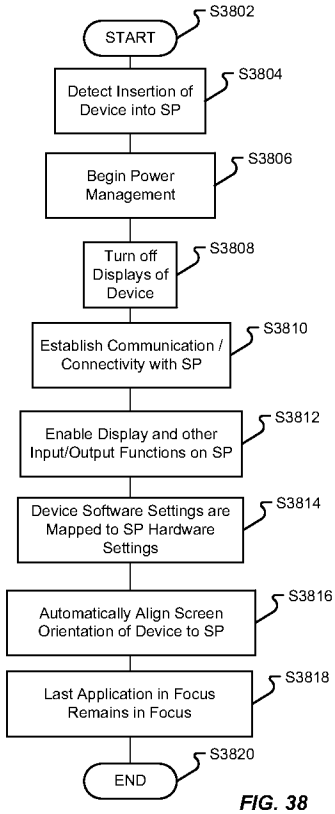


FIG. 38

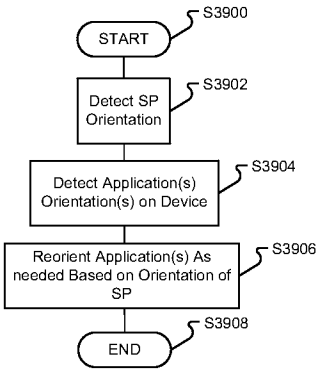


FIG. 39

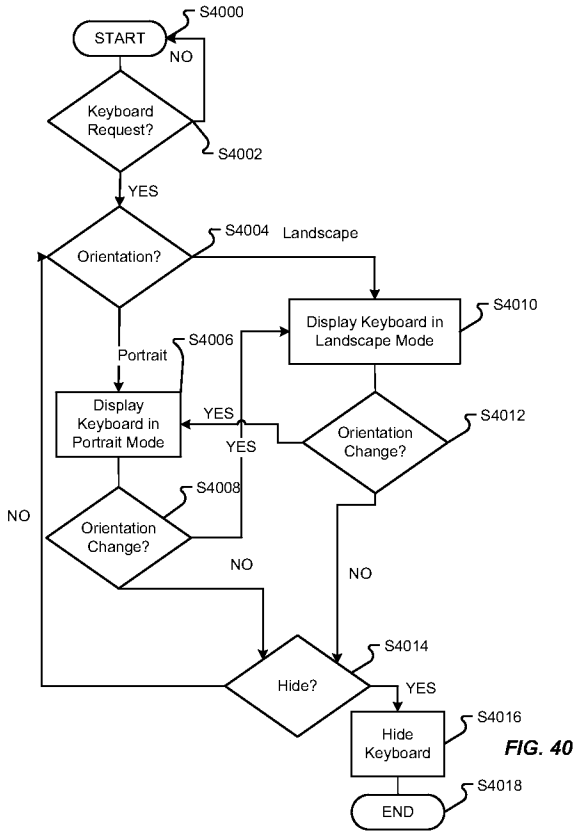


FIG. 40

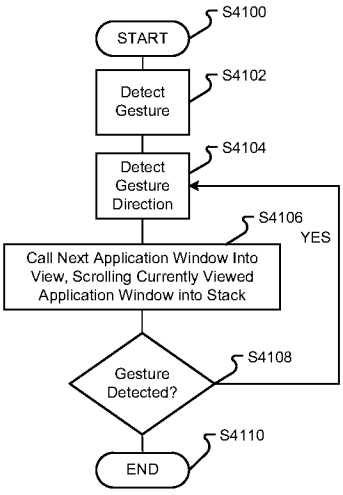


FIG. 41

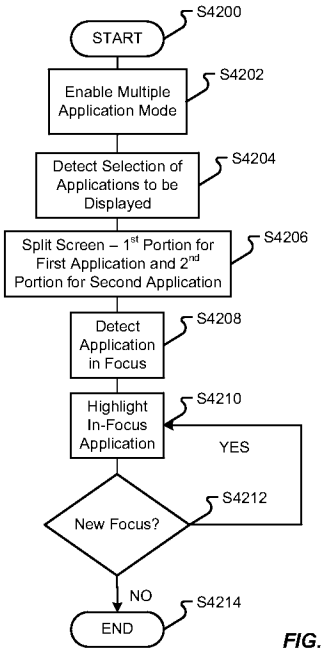


FIG. 42

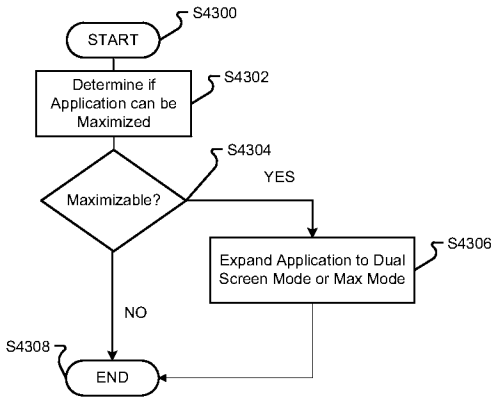


FIG. 43

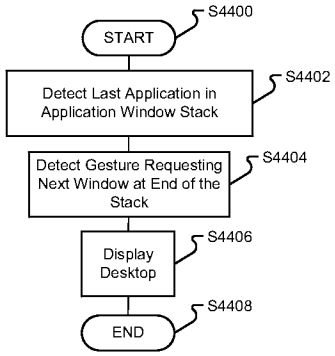


FIG. 44

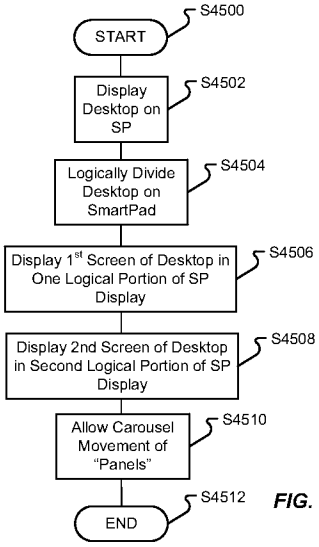


FIG. 45

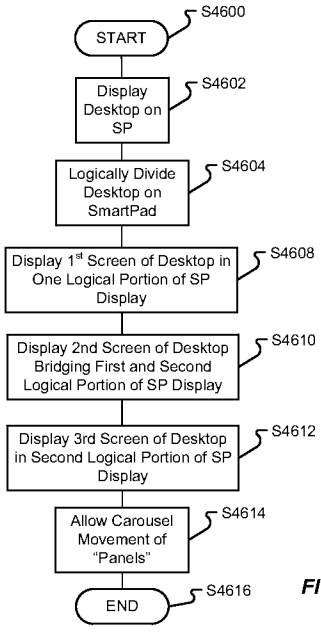


FIG. 46

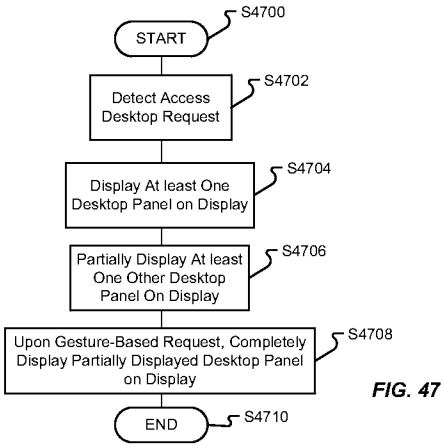

START
S4700
Detect Access Desktop Request
S4702
Display At least One Desktop Panel on Display
S4704
Partially Display At least One Other Desktop Panel On Display
S4706
Upon Gesture-Based Request, Completely Display Partially Displayed Desktop Panel on Display
S4708
END
S4710

FIG. 47

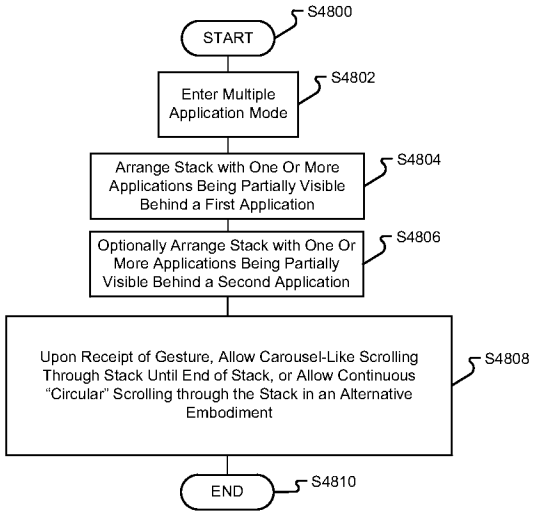

START
S4800
Enter Multiple Application Mode
S4802
Arrange Stack with One Or More Applications Being Partially Visible Behind a First Application
S4804
Optionally Arrange Stack with One Or More Applications Being Partially Visible Behind a Second Application
S4806
Upon Receipt of Gesture, Allow Carousel-Like Scrolling Through Stack Until End of Stack, or Allow Continuous "Circular" Scrolling through the Stack in an Alternative Embodiment
S4808
END
S4810

FIG. 48

SWIPEABLE KEY LINE

[0100] Fig. 7 depicts a virtual keyboard 704 that can be presented by a touch sensitive screen 104 and/or 108 provided by a device 100 in accordance with embodiments of the present disclosure. As can be appreciated by one of skill in the art after consideration of the present disclosure, the virtual keyboard 704 includes a plurality of keys or buttons 708 presented graphically by a display, 104, 108. For purposes of illustration, but without limitation, the keys or buttons 708 can include letter, number, punctuation, and control input keys. Moreover, a touch sensitive feature of the display 104, 108 allows the device 100 to receive input from a user in the form of a touch or press by the user of a button or key 708 presented by the virtual keyboard 704. In accordance with embodiments of the present disclosure, the virtual keyboard 704 can provide a plurality of user selectable virtual keysets or selectable sets of virtual keys 712. Each of the available selectable sets of virtual keys 712 can include member keys 716.

[0101] As depicted in Fig. 7, a selectable set of virtual keys 712 other than a selectable set of virtual keys 712 currently being displayed as part of a virtual keyboard 704 can be predefined and available for selection and operative use by the user. In general, the member keys 716 included in a particular selectable set of virtual keys 712 differs ftom the member keys 716 of any other selectable set of virtual keys 712, entirely or partially. For example, while a first predefined selectable set of virtual keys 712a is displayed and operative, second 712b and third 712c selectable sets of virtual keys can reside in a stack in memory 208 and can be selected for display and use in response to a user input as described herein. More particularly, in accordance with embodiments of the present invention, an alternate selectable set of virtual keys 712 can be selected by a user through input entered in an area of the virtual keyboard 704 comprising the displayed selectable set of virtual keys 712. Specifically, input in the form of a swipe, for instance a drag gesture 400 or a flick gesture 404 (See Figs. 4B and 4C) entered along a line generally corresponding or parallel to the line of keys 716 included in the displayed set of virtual keys 712 can cause the displayed selectable set of virtual keys 712 to be replaced by an alternate selectable set of virtual keys 712. More particularly, a selectable set of virtual keys 712 for display and operation can be selected from the stack of virtual keysets 712 by entering a swipe motion or gesture in a direction that is generally parallel to (*e.g.*, within ±45° of) the line along which a row of keys 708 comprising the member keys 716 of a displayed selectable set of virtual keys 112 is arranged.

[0102] In the example of Fig. 7, an active first selectable set of virtual keys 712a is displayed as part of the virtual keyboard 704. A second selectable set of virtual keys 712b and a third selectable set of virtual keys 712c are maintained in the stack and are also available, but are not currently active and are not displayed as part of the virtual keyboard 704 in the mode or state illustrated in Fig. 7. The second selectable set of virtual keys 712b can be accessed through a user input entered in an area of a touch sensitive display 100 and/or 114 that swipes the displayed selectable set of virtual keys 712a towards the right in the figure, using a drag gesture 400 or a flick gesture 404. The third selectable set of virtual keys 712c can be accessed while the first selectable set of virtual keys 712a is displayed as part of the virtual keyboard 704 by providing user input entered in an area of a touch sensitive display 110 and/or 114 swiping the displayed first selectable set of virtual keys 712a towards the left in the figure, using a drag gesture 400 or a flick gesture 404. In addition, the different selectable virtual keysets 712 can be presented as part of a virtual carousel. Accordingly, when the first selectable set of virtual keys 712a is displayed, a swiping input to the left of the figure will cause the second selectable set of virtual keys 712b to be displayed. While the second selectable set of virtual keys 712b is displayed as part of the virtual keyboard 704, a further swiping gesture of that virtual keyset 712b to the right in the figure of the second virtual keyset 712b will cause the third virtual keyset 712c to be displayed.

[0103] In accordance with embodiments of the present invention, a virtual keyboard 704 can be displayed by one or both screens 104, 108 of a dual screen device 100 as described herein. For example, in a single screen landscape mode, the virtual keyboard 704 and the active set of selectable virtual keys 712 can be displayed across a portion of one of the screens 104 or 108. As another example, in a dual screen landscape mode, the virtual keyboard 704 and the active selectable set of virtual keys 712 can be displayed across one of the screens 104 or 108, while additional application information is displayed in a second one of the screens 104 or 108. As yet another example, and as shown in Fig. 7, in a dual portrait mode, the virtual keyboard 704 and active keyset 708, including an active selectable set of virtual keys 712, may be displayed across portions of both screens 104 and 108, while other portions of the screens 104 and 108 remain available to display other information to the user. Moreover, where the active selectable set of virtual keys is displayed across portions of two screens 104, 108, a swipe (*e.g.*, a drag or flick

gesture 400 or 404) can be entered on either or both of the touch screens 104, 108 in the area in which at least a portion of the selectable set of virtual keys 712 is displayed.

[0104] Fig. 8 illustrates a virtual keyboard 704 with associated selectable sets of virtual keys 712 in a transition state. More particularly, in response to a flick or drag gesture 400 or 404 from left to right entered on a portion of the screen 104 or 108 displaying the first selectable set of virtual keys 712, that virtual keyset 712a is moved to the right in the figure, and is replaced by the second selectable set of virtual keys 712b. The third virtual keyset 712c remains available and in a stack of virtual keysets 712, but is not displayed. As depicted in the figure, the second virtual keyset 712b follows immediately after the first virtual keyset 712a, replacing the first virtual keyset 712a once the first virtual keyset 712a has been moved to the right and off of the virtual keyboard 704. Accordingly, as shown in Fig. 8, portions of multiple virtual keysets or selectable sets of virtual keys 712 can be displayed simultaneously as part of a virtual keyboard. Moreover, more than one selectable set of virtual keys 712 can, during the receipt of input ftom the user, be at least partially displayed simultaneously during the receipt of input by a user to select one of the selectable sets of virtual keys 712. In addition, a selectable set of virtual keys 712 can be displayed across at least portions of more than one screen 104, 108, where, for example, the keyboard 704 is presented by multiple screens 104, 108, for instance in a dual screen portrait display mode, while input is being entered by the user. As can be appreciated by one of skill in the art after consideration of the present disclosure, once the second selectable set of virtual keys 712b has replaced the first selectable set of virtual keys 712a, a second drag 400 or flick 404 gesture entered in the area of the screen or screens 104, 108 displaying the second selectable set of virtual keys 712b can cause the second selectable set of virtual keys 712b to be replaced by the third selectable set of virtual keys 712c. As can also be appreciated by one of skill in the art after consideration of the present disclosure, if the drag 400 or flick 404 gesture entered on the area of the display 104 and/or 108 in which the first selectable set of virtual keys 712a was displayed had been to the left in the above example, the first selectable set of virtual keys 712a would have been replaced by the third selectable set of virtual keys 712c.

[0105] In accordance with further embodiments, different numbers of selectable sets of virtual keys 712 can be provided. For example, if two selectable sets of virtual keys 712 are provided, while one of the selectable sets 712 is on display, a user can select the other of the selectable sets 712 by entering a swipe gesture on the displayed selectable set 712 in either direction. In

accordance with other embodiments, where three or more selectable sets of virtual keys 712 are available, no more than portions of two adjacent selectable sets of virtual keys 712 may be presented as part of the keyboard 704 at any one point in time. The other selectable sets of virtual keys 712 can be maintained in a stack in memory, and selected through the successive entry of swipe gestures by the user, until the desired selectable set of virtual keys 712 has been reached and is displayed as part of the virtual keyboard 704.

[0106] With reference now to Fig. 9, aspects of a method for providing swipeable virtual keysets 708 that include swipeable or selectable sets of virtual keys 712 in accordance with embodiments of the present invention are depicted. Initially, the system is started, and a determination is made as to whether a virtual keyboard 704 is to be displayed (step 904). The system idles at step 904 until a decision to display a virtual keyboard 704 is reached. A virtual keyboard 704, including a selectable set of virtual keys 712, is displayed at step 908 after a determination to display the virtual keyboard 704 has been made. As can be appreciated by one of skill in the art after consideration of the present disclosure, a virtual keyboard 704 may be displayed in response to determining that input ftom a user is required or indicated. For example, a user may make a selection or otherwise enter a command through an input of the device 100 to display a virtual keyboard 704, or that implies a need to present a virtual keyboard 704, for example by selecting a text entry box provided as part of a graphical user interface. In accordance with still other embodiments, a need to display a virtual keyboard 704 can be determined through the operation of application programming. For example, a virtual keyboard 704 can be displayed when an application being executed by or in association with the device 100 reaches a point at which user input is required.

[0107] At step 912, a determination is made as to whether input to change the displayed selectable set of virtual keys 712 has been received. If input to change the displayed set of virtual keys has been received, a next selectable set of virtual keys 712 is identified (step 916). The identification of the next selectable set of virtual keys 712 to display can be made in view of various considerations. These include the number of selectable sets of virtual keys 712 available, their relative location in a stack of selectable sets of virtual keys 712 maintained in memory 208, 508, and the direction of a swiping motion received as the input from the user. Once the next selectable set of virtual keys 712 has been identified, the process returns to step 908, with the display of the identified selectable set of virtual keys 712.

[0108] If input to change the displayed selectable set of virtual keys 712 is not received at step 912, a determination can be made as to whether input to discontinue a display of a selectable set of virtual keys 712 has been received (step 920). If input to discontinue the display of the selectable set of virtual keys 712 has been received, a revised virtual keyboard 704 can be displayed (step 924). Input to discontinue a display of a selectable set of virtual keys 712 can include the selection of a menu item, or a change in the device 100 operating mode. For example, through the receipt of input from a user or the execution of instructions as part of the operation of an application, the display of a selectable set of virtual keys 712 may be discontinued, for instance because the included member keys 716 are no longer required, because the included member keys 716 are no longer relevant to the operating mode or context of the device 100, or for other reasons.

[0109] If at step 920 input to discontinue the display of a selectable set of virtual keys 712 is not received, a determination can next be made as to whether to continue to display the virtual keyboard 704 (step 928). For example, user input or the operation or execution of application programming can signal that the display of the virtual keyboard 704 should be discontinued. If the display is to be continued, the process can return to step 904. Alternatively, if the display of the virtual keyboard 704 is to be discontinued, the process may end.

[0110] In accordance with embodiments of the present invention, a virtual keyboard 704 including the selectable set of virtual keys 712 feature described herein can be implemented by application programming or instructions executed by an associated device 100. For example, the virtual keyboard 704 and the provision of selectable sets of virtual keys 712 can be provided by or in association with one or more applications stored in memory 208, 508 and executed by a processor 204, 504. Moreover, the provision of a virtual keyboard 704 can be in association with screens 104, 108 including touch sensitive display 110, 114 regions and the operation of one or more associated display controllers 216. Moreover, the provision of a virtual keyboard 704 in accordance with embodiments of the present invention can be in view of inputs from various applications and components of the device 100. For instance, a basic virtual keyboard function can be provided by the operating system 516 in association with an operating system 516 and/or framework 520. Moreover, the available selectable sets of virtual keys 712 can be determined, at least in part, by a particular application 564 being executed and for which user input is provided using the virtual keyboard 704.

[0111] The exemplary systems and methods of this disclosure have been described in relation to swipeable key lines or selectable sets of virtual keys. However, to avoid unnecessarily obscuring the present disclosure, the preceding description omits a number of known structures and devices. This omission is not to be construed as a limitation of the scopes of the claims. Specific details are set forth to provide an understanding of the present disclosure. It should however be appreciated that the present disclosure may be practiced in a variety of ways beyond the specific detail set forth herein.

[0112] Furthermore, while the exemplary aspects, embodiments, and/or configurations illustrated herein show the various components of the system collocated, certain components of the system can be located remotely, at distant portions of a distributed network, such as a LAN and/or the Internet, or within a dedicated system. Thus, it should be appreciated, that the components of the system can be combined in to one or more devices, such as a computer, laptop, netbook, tablet computer, smart phone, mobile device, etc., or collocated on a particular node of a distributed network, such as an analog and/or digital telecommunications network, a packet-switch network, or a circuit-switched network. It will be appreciated from the preceding description, and for reasons of computational efficiency, that the components of the system can be arranged at any location within a distributed network of components without affecting the operation of the system. For example, the various components can be located in a switch such as a PBX and media server, gateway, in one or more communications devices, at one or more users' premises, or some combination thereof. Similarly, one or more functional portions of the system could be distributed between a telecommunications device(s) and an associated computing device.

[0113] Furthermore, it should be appreciated that the various links connecting the elements can be wired or wireless links, or any combination thereof, or any other known or later developed element(s) that is capable of supplying and/or communicating data to and from the connected elements. These wired or wireless links can also be secure links and may be capable of communicating encrypted information. Transmission media used as links, for example, can be any suitable carrier for electrical signals, including coaxial cables, copper wire and fiber optics, and may take the form of acoustic or light waves, such as those generated during radio-wave and infra-red data communications.

[0114] Also, while the flowcharts have been discussed and illustrated in relation to a particular sequence of events, it should be appreciated that changes, additions, and omissions to this sequence can occur without materially affecting the operation of the disclosed embodiments, configuration, and aspects.

[0115] A number of variations and modifications of the disclosure can be used. It would be possible to provide for some features of the disclosure without providing others.

[0116] For example in one alternative embodiment, the entirety of a virtual keyboard 704 is comprised of a swipeable key line or selectable set of virtual keys 712.

[0117] In another alternative embodiment, the selectable set of virtual keys 712 are presented as a two-dimensional array of keys 716, rather than a line.

[0118] In yet another embodiment, the systems and methods of this disclosure can be implemented in conjunction with a special purpose computer, a programmed microprocessor or microcontroller and peripheral integrated circuit element(s), an ASIC or other integrated circuit, a digital signal processor, a hard-wired electronic or logic circuit such as discrete element circuit, a programmable logic device or gate array such as PLD, PLA, FPGA, PAL, special purpose computer, any comparable means, or the like. In general, any device(s) or means capable of implementing the methodology illustrated herein can be used to implement the various aspects of this disclosure. Exemplary hardware that can be used for the disclosed embodiments, configurations and aspects includes computers, handheld devices, telephones (e.g., cellular, Internet enabled, digital, analog, hybrids, and others), and other hardware known in the art. Some of these devices include processors (e.g., a single or multiple microprocessors), memory, nonvolatile storage, input devices, and output devices. Furthermore, alternative software implementations including, but not limited to, distributed processing or component/object distributed processing, parallel processing, or virtual machine processing can also be constructed to implement the methods described herein.

[0119] In yet another embodiment, the disclosed methods may be readily implemented in conjunction with software using object or object-oriented software development environments that provide portable source code that can be used on a variety of computer or workstation platforms. Alternatively, the disclosed system may be implemented partially or fully in hardware using standard logic circuits or VLSI design. Whether software or hardware is used to implement the systems in accordance with this disclosure is dependent on the speed and/or

efficiency requirements of the system, the particular function, and the particular software or hardware systems or microprocessor or microcomputer systems being utilized.

[0120] In yet another embodiment, the disclosed methods may be partially implemented in software that can be stored on a storage medium, executed on programmed general-purpose computer with the cooperation of a controller and memory, a special purpose computer, a microprocessor, or the like. In these instances, the systems and methods of this disclosure can be implemented as program embedded on personal computer such as an applet, JAVA® or CGI script, as a resource residing on a server or computer workstation, as a routine embedded in a dedicated measurement system, system component, or the like. The system can also be implemented by physically incorporating the system and/or method into a software and/or hardware system.

[0121] Although the present disclosure describes components and functions implemented in the aspects, embodiments, and/or configurations with reference to particular standards and protocols, the aspects, embodiments, and/or configurations are not limited to such standards and protocols. Other similar standards and protocols not mentioned herein are in existence and are considered to be included in the present disclosure. Moreover, the standards and protocols mentioned herein and other similar standards and protocols not mentioned herein are periodically superseded by faster or more effective equivalents having essentially the same functions. Such replacement standards and protocols having the same functions are considered equivalents included in the present disclosure.

[0122] The present disclosure, in various aspects, embodiments, and/or configurations, includes components, methods, processes, systems and/or apparatus substantially as depicted and described herein, including various aspects, embodiments, configurations embodiments, subcombinations, and/or subsets thereof. Those of skill in the art will understand how to make and use the disclosed aspects, embodiments, and/or configurations afrer understanding the present disclosure. The present disclosure, in various aspects, embodiments, and/or configurations, includes providing devices and processes in the absence of items not depicted and/or described herein or in various aspects, embodiments, and/or configurations hereof, including in the absence of such items as may have been used in previous devices or processes, e.g., for improving performance, achieving ease and\or reducing cost of implementation.

[0123] The foregoing discussion has been presented for purposes of illustration and description. The foregoing is not intended to limit the disclosure to the form or forms disclosed herein. In the foregoing Detailed Description for example, various features of the disclosure are grouped together in one or more aspects, embodiments, and/or configurations for the purpose of streamlining the disclosure. The features of the aspects, embodiments, and/or configurations of the disclosure may be combined in alternate aspects, embodiments, and/or configurations other than those discussed above. This method of disclosure is not to be interpreted as reflecting an intention that the claims require more features than are expressly recited in each claim. Rather, as the following claims reflect, inventive aspects lie in less than all features of a single foregoing disclosed aspect, embodiment, and/or configuration. Thus, the following claims are hereby incorporated into this Detailed Description, with each claim standing on its own as a separate preferred embodiment of the disclosure.

[0124] Moreover, though the description has included description of one or more aspects, embodiments, and/or configurations and certain variations and modifications, other variations, combinations, and modifications are within the scope of the disclosure, e.g., as may be within the skill and knowledge of those in the art, after understanding the present disclosure. It is intended to obtain rights which include alternative aspects, embodiments, and/or configurations to the extent permitted, including alternate, interchangeable and/or equivalent structures, functions, ranges or steps to those claimed, whether or not such alternate, interchangeable and/or equivalent structures, functions, ranges or steps are disclosed herein, and without intending to publicly dedicate any patentable subject matter.

What is claimed is:

1. A method for providing an input device, comprising:

presenting by a touch screen display of a first device a first selectable set of virtual keys;

receiving a first input from a user, wherein the first input includes a touch input in an area of the touch screen display corresponding to the displayed first selectable set of virtual keys;

in response to the received first input, presenting by the touch screen display a second selectable set of virtual keys.

2. The method of Claim 1, wherein the first input includes a gesture.

3. The method of Claim 2, wherein the gesture includes a swipe gesture.

4. The method of Claim 1, wherein the touch screen display includes first and second touch screen displays, wherein a first portion of the first selectable set of virtual keys is presented by the first touch screen display during a first time interval, wherein a second portion of the first selectable set of virtual keys is presented by the second touch screen display during the first time interval, wherein a first portion of the second selectable set of virtual keys is presented by the first touch screen display during a second time interval, wherein a second portion of the second selectable set of virtual keys is presented by the second touch screen display during the second time interval, wherein the second time interval is afrer the first time interval, and wherein at least some portion of the second time interval is afrer the first input is received from the user.

5. The method of Claim 1, wherein the first selectable set of virtual keys is presented as a row of virtual keys, wherein the second selectable set of virtual keys is presented as a row of virtual keys.

6. The method of Claim 5, wherein a portion of the first selectable set of virtual keys and a portion of the second selectable set of virtual keys are simultaneously presented during at least a first time interval corresponding to the receipt of the first input from the user.

7. The method of Claim 1, further comprising receiving a second input from the user, wherein the second input includes a touch input in an area of the first touch screen corresponding to the displayed second selectable set of virtual keys.

8. The method of Claim 7, wherein in response to the received second input one of the first selectable set of virtual keys and a third selectable set of virtual keys is presented.

9. The method of Claim 8, wherein the first selectable set of virtual keys is presented as a row of virtual keys, wherein the second selectable set of virtual keys is presented as a row of virtual keys, wherein the third selectable set of virtual keys is presented as a row of virtual keys, wherein the first input is in a first direction along the row of virtual keys of the first selectable set of virtual keys and causes the second selectable set of virtual keys to be presented, wherein the second input is in the first direction along the row of virtual keys of the second selectable set of virtual keys and causes the third selectable set of virtual keys to be presented.

10. The method of Claim 9, further comprising receiving third input from the user, wherein the third input includes a touch input in an area of the first touch screen display corresponding to the displayed third selectable set of virtual keys, wherein the third input is in the first direction along the row of virtual keys of the third selectable set of virtual keys and causes the first selectable set of virtual keys to be presented.

11. A device, comprising:

a touch screen display;

memory;

a processor;

application programming stored in the memory and executed by the processor, wherein the application programming is operable to:

display a first selectable set of virtual keys;

receive first touch screen input entered in an area including the displayed first selectable set of virtual keys;

in response to the received first touch screen input, display a second selectable set of virtual keys.

12. The device of Claim 11, further comprising:

a first touch screen display;

a second touch screen display, wherein the first and second touch screen displays are operated by the application programming to display the first selectable set of virtual keys during a first interval of time, and wherein the first and second touch screen displays are operated by the application programming to display the second selectable set of virtual keys during a second interval of time.

13. The device of Claim 12, wherein the first selectable set of virtual keys is displayed as a row of virtual keys extending across at least a portion of the first touch screen display and across at least a portion of the second touch screen display, wherein the first touch screen input is received by at least one of the first and second touch screen displays.

14. The device of Claim 13, wherein the first touch screen input is in a first direction along the row of virtual keys, wherein the application programming is further operable to receive second touch screen input in an area including at least a portion of the second selectable set of virtual keys, wherein the second touch screen input is in the first direction along the row of virtual keys, wherein in response to the received second touch screen input the application programming is operable to display a third selectable set of virtual keys.

15. The device of Claim 13, wherein the first touch screen input is in a first direction along the row of virtual keys, wherein the application programming is further operable to receive second touch screen input in an area including at least a portion of the second selectable set of virtual keys, wherein the second touch screen input is in a second direction along the row of virtual keys, wherein in response to the received second touch screen input the application programming is operable to display first selectable set of virtual keys.

16. A computer readable medium having stored thereon computer-executable instructions, the computer executable instructions causing a processor to execute a method for displaying selectable virtual key sets, the computer executable instructions comprising:

instructions to display a first set of virtual keys on a touch screen;

instructions to determine whether at least a first touch screen input has been received in an area of the touch screen corresponding to the displayed first selectable set of virtual keys;

instructions to display a second selectable set of virtual keys on the touch screen in place of the first selectable set of virtual keys in response to the at least a first touch screen input.

17. The computer readable medium of Claim 16, wherein the first selectable set of virtual keys is displayed as a row of virtual keys, wherein the at least a first touch screen input includes a swipe gesture along the first selectable set of virtual keys.

18. The computer readable medium of Claim 17, wherein during at least a portion of receiving the first touch screen input a portion of the first selectable set of virtual keys and a portion of the second selectable set of virtual keys are displayed simultaneously.

19. The computer readable medium of Claim 18, the computer executable instructions further comprising:

instructions to display the first selectable set of virtual keys on a touch screen that includes first and second touch screens during a first time interval, wherein a first portion of the first selectable set of virtual keys is displayed on the first touch screen and a second portion of the first selectable set of virtual keys is displayed on the second touch screen during the first time interval;

instructions to display the second selectable set of virtual keys on the touch screen that includes first and second touch screens during a second time interval, wherein a first portion of the second selectable set of virtual keys is displayed on the first touch screen and a second portion of the second selectable set of virtual keys is displayed on the second touch screen during the second time interval, wherein the first time interval is before the first touch screen input is received, and wherein the second time interval is after the first touch screen input is received.

20. The computer readable medium of Claim 19, the computer readable instructions further comprising:

instructions to display at least some of the second portion of the first selectable set of virtual keys on the first touch screen display while the first touch screen input is being received;

instructions to display at least some of the first portion of the second selectable set of virtual keys on the second touch screen display while the first touch screen input is being received.

ABSTRACT

Methods and devices for the display of virtual keyboards are provided. More particularly, virtual keyboards that include selectable sets of virtual keys that can be selected or changed through direct input by a user are provided. The user input can include a touch screen input or gesture applied to an area of a display in which a selectable set of virtual keys is presented. The touch screen input or swipe can cause the displayed selectable set of virtual keys to be replaced by an alternate selectable set of virtual keys.

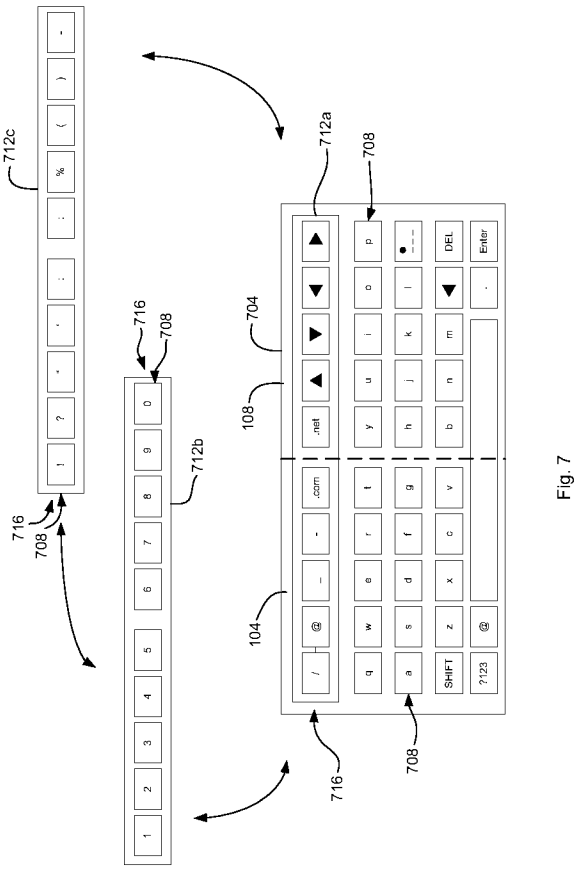


Fig. 7

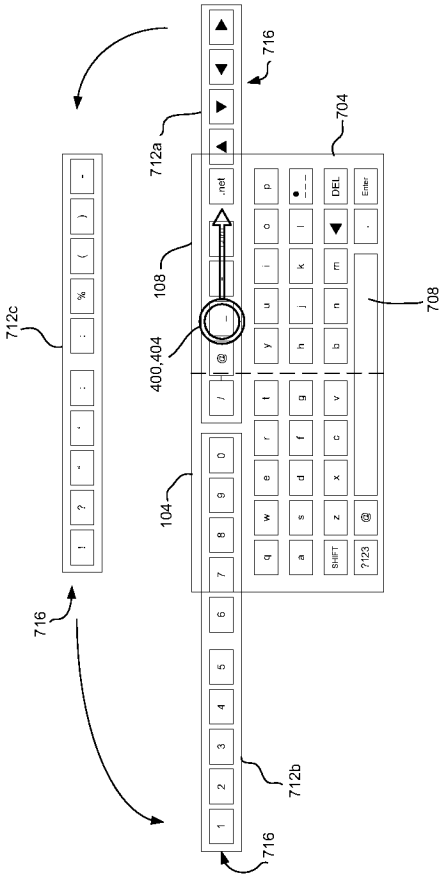


Fig. 8

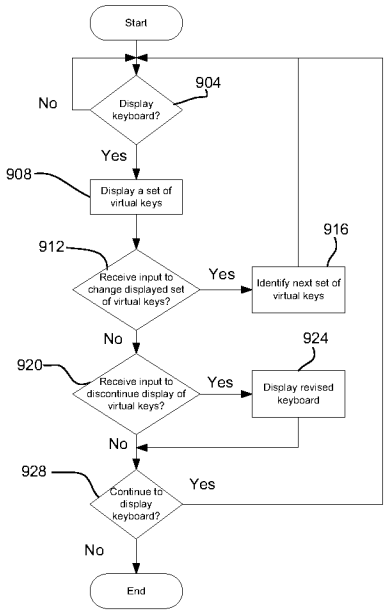


Fig. 9

DISPLAYED IMAGE TRANSITION INDICATOR

[0100] The device 100 manages desktops and/or windows with at least one window stack 1700, 1728, as shown in Figs. 10A and 10B. A window stack 1700, 1728 is a logical arrangement of active and/or inactive windows for a multi-screen device. For example, the window stack 1700, 1728 may be logically similar to a deck of cards, where one or more windows or desktops are arranged in order, as shown in Figs. 10A and 10B. An active window is a window that is currently being displayed on at least one of the touch sensitive displays 110, 114. For example, windows 104 and 108 are active windows and are displayed on touch sensitive displays 110 and 114. An inactive window is a window that was opened and displayed but is now "behind" an active window and not being displayed. In embodiments, an inactive window may be for an application that is suspended, and thus, the window is not displaying active content. For example, windows 1712, 1716, 1720, and 1724 are inactive windows.

[0101] A window stack 1700, 1728 may have various arrangements or organizational structures. In the embodiment shown in Fig. 10A, the device 100 includes a first stack 1760 associated with a first touch sensitive display 110 and a second stack associated with a second touch sensitive display 114. Thus, each touch sensitive display 110, 114 can have an associated window stack 1760, 1764. These two window stacks 1760, 1764 may have different numbers of windows arranged in the respective stacks 1760, 1764. Further, the two window stacks 1760, 1764 can also be identified differently and managed separately. Thus, the first window stack 1760 can be arranged in order from a first window 1704 to a next window 1720 to a last window 1724 and finally to a desktop 1722, which, in embodiments, is at the "bottom" of the window stack 1760. In embodiments, the desktop 1722 is not always at the "bottom" as application windows can be arranged in the window stack below the desktop 1722, and the desktop 1722 can be brought to the "top" of a stack over other windows during a desktop reveal. Likewise, the second stack 1764 can be arranged from a first window 1708 to a next window 1712 to a last window 1716, and finally to a desktop 1718, which, in embodiments, is a single desktop area, with desktop 1722, under all the windows in both window stack 1760 and window stack 1764. A logical data structure for managing the two window stacks 1760, 1764 may be as described in conjunction with Fig. 11.

[0102] Another arrangement for a window stack 1728 is shown in Fig. 10B. In this embodiment, there is a single window stack 1728 for both touch sensitive displays 110, 114.

a function of the orientation of the device 100, the context of what programs, functions, software, etc. are being executed on the device 100, how the stack is positioned when the new window is opened, etc. To insert the window in the stack, the position in the stack for the window is determined and the touch sensitive display 110, 114 to which the window is associated may also be determined. With this information, a logical data structure for the window can be created and stored. When user interface or other events or tasks change the arrangement of windows, the window stack(s) can be changed to reflect the change in arrangement. It should be noted that these same concepts described above can be used to manage the one or more desktops for the device 100.

[0106] A logical data structure 1800 for managing the arrangement of windows or desktops in a window stack is shown in Fig. 11. The logical data structure 1800 can be any data structure used to store data whether an object, record, file, etc. The logical data structure 1800 can be stored in any type of database or data storage system, regardless of protocol or standard. In embodiments, the logical data structure 1800 includes one or more portions, fields, attributes, etc. that store data in a logical arrangement that allows for easy storage and retrieval of the information. Hereinafrer, these one or more portions, fields, attributes, etc. shall be described simply as fields. The fields can store data for a window identifier 1804, dimensions 1808, a stack position identifier 1812, a display identifier 1816, and/or an active indicator 1820. Each window in a window stack can have an associated logical data structure 1800. While only a single logical data structure 1800 is shown in Fig. 11, there may be more or fewer logical data structures 1800 used with a window stack (based on the number of windows or desktops in the stack), as represented by ellipses 1824. Further, there may be more or fewer fields than those shown in Fig. 11, as represented by ellipses 1828.

[0107] A window identifier 1804 can include any identifier (ID) that uniquely identifies the associated window in relation to other windows in the window stack. The window identifier 1804 can be a globally unique identifier (GUID), a numeric ID, an alphanumeric ID, or other type of identifier. In embodiments, the window identifier 1804 can be one, two, or any number of digits based on the number of windows that can be opened. In alternative embodiments, the size of the window identifier 1804 may change based on the number of windows opened. While the window is open, the window identifier 1804 may be static and remain unchanged.

Thus, the window stack 1728 is arranged from a desktop 1758 to a first window 1744 to a last window 1756. A window can be arranged in a position among all windows without an association to a specific touch sensitive display 110, 114. In this embodiment, a window is in the order of windows. Further, at least one window is identified as being active. For example, a single window may be rendered in two portions 1732 and 1736 that are displayed on the first touch sensitive screen 110 and the second touch sensitive screen 114. The single window may only occupy a single position in the window stack 1728 although it is displayed on both displays 110, 114.

[0103] Yet another arrangement of a window stack 1760 is shown in Figs. 10C through 10E. The window stack 1760 is shown in three "elevation" views. In Fig. 10C, the top of the window stack 1760 is shown. Two sides of the window stack 1760 are shown in Figs. 10D and 10E. In this embodiment, the window stack 1760 resembles a stack of bricks. The windows are stacked on each other. Looking from the top of the window stack 1760 in Fig, 10C, only the top most windows in the window stack 1760 are seen in different portions of the composite display 1764. The composite display 1764 represents a logical model for the entire display area of the device 100, which can include touch sensitive display 110 and touch sensitive display 114. A desktop 1786 or a window can occupy part or all of the composite display 1764.

[0104] In the embodiment shown, the desktop 1786 is the lowest display or "brick" in the window stack 1760. Thereupon, window 1 1782, window 2 1782, window 3 1768, and window 4 1770 are layered. Window 1 1782, window 3 1768, window 2 1782, and window 4 1770 only occupy a portion of the composite display 1764. Thus, another part of the stack 1760 includes window 8 1774 and windows 5 through 7 shown in section 1790. Only the top window in any portion of the composite display 1764 is actually rendered and displayed. Thus, as shown in the top view in Fig. 10C, window 4 1770, window 8 1774, and window 3 1768 are displayed as being at the top of the display in different portions of the window stack 1760. A window can be dimensioned to occupy only a portion of the composite display 1760 to "reveal" windows lower in the window stack 1760. For example, window 3 1768 is lower in the stack than both window 4 1770 and window 8 1774 but is still displayed. A logical data structure to manage the window stack can be as described in conjunction with Fig. 11.

[0105] When a new window is opened, the newly activated window is generally positioned at the top of the stack. However, where and how the window is positioned within the stack can be

[0108] Dimensions 1808 can include dimensions for a window in the composite display 1760. For example, the dimensions 1808 can include coordinates for two or more corners of the window or may include one coordinate and dimensions for the width and height of the window. These dimensions 1808 can delineate what portion of the composite display 1760 the window may occupy, which may the entire composite display 1760 or only part of composite display 1760. For example, window 4 1770 may have dimensions 1880 that indicate that the window 1770 will occupy only part of the display area for composite display 1760, as shown in Figs. 10C through 10E. As windows are moved or inserted in the window stack, the dimensions 1808 may change.

[0109] A stack position identifier 1812 can be any identifier that can identify the position in the stack for the window or may be inferred from the window's control record within a data structure, such as a list or a stack. The stack position identifier 1812 can be a GUID, a numeric ID, an alphanumeric ID, or other type of identifier. Each window or desktop can include a stack position identifier 1812. For example, as shown in Fig. 10A, window 1 1704 in stack 1 1760 can have a stack position identifier 1812 of 1 identifying that window 1704 is the first window in the stack 1760 and the active window. Similarly, window 6 1724 can have a stack position identifier 1812 of 3 representing that window 1724 is the third window in the stack 1760. Window 2 1708 can also have a stack position identifier 1812 of 1 representing that window 1708 is the first window in the second stack 1764. As shown in Fig. 10B, window 1 1744 can have a stack position identifier 1812 of 1, window 3, rendered in portions 1732 and 1736, can have a stack position identifier 1812 of 3, and window 6 1756 can have a stack position identifier 1812 of 6. Thus, depending on the type of stack, the stack position identifier 1812 can represent a window's location in the stack.

[0110] A display identifier 1816 can identify that the window or desktop is associated with a particular display, such as the first display 110 or the second display 114, or the composite display 1760 composed of both displays. While this display identifier 1816 may not be needed for a multi-stack system, as shown in Fig. 10A, the display identifier 1816 can indicate whether a window in the serial stack of Fig. 10B is displayed on a particular display. Thus, window 3 may have two portions 1732 and 1736 in Fig. 10B. The first portion 1732 may have a display identifier 1816 for the first display while the second portion 1736 may have a display identifier 1816 for the second display 114. However, in alternative embodiments, the window may have

two display identifier 1816 that represent that the window is displayed on both of the displays 110, 114, or a display identifier 1816 identifying the composite display. In another alternate embodiment, the window may have a single display identifier 1816 to represent that the window is displayed on both of the displays 110, 114.

[0111] Similar to the display identifier 1816, an active indicator 1820 may not be needed with the dual stack system of Fig. 10A, as the window in stack position 1 is active and displayed. In the system of Fig. 10B, the active indicator 1820 can indicate which window(s) in the stack is being displayed. Thus, window 3 may have two portions 1732 and 1736 in Fig. 10. The first portion 1732 may have an active indicator 1820 while the second portion 1736 may also have an active indicator 1820. However, in alternative embodiments, window 3 may have a single active indicator 1820. The active indicator 1820 can be a simple flag or bit that represents that the window is active or displayed.

[0112] An embodiment of a method 1900 for creating a window stack is shown in Fig. 12. While a general order for the steps of the method 1900 is shown in Fig. 12. Generally, the method 1900 starts with a start operation 1904 and ends with an end operation 1928. The method 1900 can include more or fewer steps or can arrange the order of the steps differently than those shown in Fig. 12. The method 1900 can be executed as a set of computer-executable instructions executed by a computer system and encoded or stored on a computer readable medium. Hereinafrer, the method 1900 shall be explained with reference to the systems, components, modules, software, data structures, user interfaces, etc. described in conjunction with Figs. 1-11.

[0113] A multi-screen device 100 can receive activation of a window, in step 1908. In embodiments, the multi-screen device 100 can receive activation of a window by receiving an input from the touch sensitive display 110 or 114, the configurable area 112 or 116, a gesture capture region 120 or 124, or some other hardware sensor operable to receive user interface inputs. The processor may execute the Task Management Module 540 may receive the input. The Task Management Module 540 can interpret the input as requesting an application task to be executed that will open a window in the window stack.

[0114] In embodiments, the Task Management Module 540 places the user interface interaction in the task stack 552 to be acted upon by the Display Configuration Module 568 of the Multi-Display Management Module 524. Further, the Task Management Module 540 waits

for information from the Multi-Display Management Module 524 to send instructions to the Window Management Module 532 to create the window in the window stack.

[0115] The Multi-Display Management Module 524, upon receiving instruction from the Task Management Module 540, determines to which touch portion of the composite display 1760, the newly activated window should be associated, in step 1912. For example, window 4 1770 is associated with the a portion of the composite display 1764 In embodiments, the device state module 574 of the Multi-Display Management Module 524 may determine how the device is oriented or in what state the device is in, e.g., open, closed, portrait, etc. Further, the preferences module 572 and/or requirements module 580 may determine how the window is to be displayed. The gesture module 576 may determine the user's intentions about how the window is to be opened based on the type of gesture and the location of where the gesture is made.

[0116] The Display Configuration Module 568 may use the input from these modules and evaluate the current window stack 1760 to determine the best place and the best dimensions, based on a visibility algorithm, to open the window. Thus, the Display Configuration Module 568 determines the best place to put the window at the top of the window stack 1760, in step 1916. The visibility algorithm, in embodiments, determines for all portions of the composite display, which windows are at the top of the stack. For example, the visibility algorithm determines that window 3 1768, window 4 1770, and window 8 1774 are at the top of the stack 1760 as viewed in Figs. 10C through 10E. Upon determining where to open the window, the Display Configuration Module 568 can assign a display identifier 816 and possibly dimensions 808 to the window. The display identifier 816 and dimensions 808 can then be sent back to the Task Management Module 540. The Task Management Module 540 may then assign the window a stack position identifier 812 indicating the windows position at the top of the window stack.

[0117] In embodiments, the Task Management Module 540 sends the window stack information and instructions to render the window to the Window Management Module 532. The Window Management Module 532 and the Task Management Module 540 can create the logical data structure 800, in step 1924. Both the Task Management Module 540 and the Window Management Module 532 may create and manage copies of the window stack. These copies of the window stack can be synchronized or kept similar through communications between the Window Management Module 532 and the Task Management Module 540. Thus,

the Window Management Module 532 and the Task Management Module 540, based on the information determined by the Multi-Display Management Module 524, can assign dimensions 808, a stack position identifier 812 (e.g., window 1 1782, window 4 1770, etc.), a display identifier 816 (e.g., touch sensitive display 1 110, touch sensitive display 2 114, composite display identifier, etc,), and an active indicator 820, which is generally always set when the window is at the "top" of the stack. The logical data structure 800 may then be stored by both the Window Management Module 532 and the Task Management Module 540. Further, the Window Management Module 532 and the Task Management Module 540may thereinafrer manage the window stack and the logical data structure(s) 800.

[0001] FIG. 13 depicts a further window stacking configuration. A plurality of windows 1, 2, 3, 4, 5, 6, 7, and 8, whether from the same or different multi-screen or single-screen applications, are depicted. The touch sensitive display 110 currently has window 4 in the active display position while touch sensitive display 114 currently has window 5 in the active display position. The stack for touch sensitive display 110, from top to bottom, has window 4 in the active display position and windows 3, 2, and 1 positioned behind it. The stack for touch sensitive display 114, from top to bottom, has window 5 in the active display position and windows 6, 7, and 8 positioned behind it.

[0118] The desktops D1, D2, D3, D4, D5, and D6 are positioned behind the window stacks. The desktops can be viewed as a desktop stack different from the window stack. Viewed in this manner, the touch sensitive display 110 has a corresponding desktop stack comprising desktops D3, D2, and D1,with desktop D1 being at the bottom 2300 stack position and with desktop D3 at the top stack position being capable of being displayed with window 4 (depending on the window position and size (whether maximized or minimized), and the touch sensitive display 114 has a corresponding desktop stack has a corresponding desktop stack comprising desktops D4, D5, and D6,with desktop D6 being at the bottom 2304 stack position and with desktop D4 at the top stack position being capable of being displayed with window 5 (depending on the window position and size (whether maximized or minimized). Conceptually, the desktops can be viewed as a canvas divided, in this example, into six sections of which two can be displayed, at any one time, the touch sensitive displays 110, 114. When the device 100 is in the closed state, this conceptual model, in one configuration, persists. In this configuration, only one window and desktop stacks can be seen (that corresponding to the primary screen) but the other

window and desktop stacks are virtual; that is, they are maintained in memory but cannot be seen because the secondary screen is not enabled.

[0119] A displayed image transition indicator, also referred to as a well, is displayed ahead of the displayed image (such as a window or desktop) as the displayed image moves from an origin to a target touch sensitive display 110, 114. The displayed image transition indicator previews for the user movement of the displayed image in response to a user gesture. For example when a window movement gesture is received from the user, the transition indicator expands or slides from behind the window (to be moved) and moves along the intended path of travel for the window, or towards the target touch sensitive display, to the final window destination. The portion of the target touch sensitive display to be occupied by the window after movement is occupied by the transition indicator. Afrer movement of the transition indicator is completed or at some other point of transition indicator movement, the window moves to occupy the portion of the target touch sensitive display occupied by the transition indicator. In one configuration, the displayed image transition indicator moves at substantially the same rate as (or as a linear or other function of a rate of movement of) the tracked user gesture causing movement of the displayed object. The displayed image transition indicator, in configuration, is used for multi-screen applications that need to expand to two screens or touch sensitive displays. In this configuration, the displayed image transition indicator is not used for moving displayed images associated with single-screen applications. In that case, the displayed image and output of the actual application are moved without a transition indicator being involved.

[0120] The transition indicator is typically a displayed image, commonly of a similar size and shape to the displayed image to be moved, and is unable to receive or provide dynamic user input or output, respectively. It typically, though not necessarily, is a substantially solid colored displayed image that has an appearance different from the displayed image to be moved. The transition indicator may display a trademark or other brand image associated with a manufacturer, wholesaler, or retailer of the device 100. In other configurations, the transition indicator is user configurable. The user may select a color or set of colors, pattern, design, icon, photograph or other image to be displayed as the transition indicator. Thus, the user may personalize the transition indicator to suit his or her preferences, whereby different users have different transition indicators on their respective devices 100. The user may further select one or more audible sounds to be played at one or more selected points in the transition indicator's

travel to the target touch sensitive display. For example, a user selected sound may be played to announce initiation of movement of the transition indicator, at the half-way point along the transition indicator's path of travel, and/or when the transition indicator is at its destination in the target touch sensitive display. The user may further elect to disable the transition indicator, resize the transition indicator so that its size is less than or larger than the displayed image to be relocated, and/or re-time the movement of the transition indicator to make its movement faster or slower than a default setting.

[0121] In one configuration, the well is an affordance generally used for multi-screen applications and not for single-screen applications. In one configuration, the transition indicator is enabled only when displayed images are to be moved in response to a gesture received by a gesture capture region 120, 124. In one configuration, the transition indicator is enabled only when the displayed images are to be moved in response to a gesture received by a touch sensitive display 110, 114. In other configurations, the transition indicator is associated only with certain displayed image movements or transitions.

[0122] Various examples will now be discussed with reference to FIGS. 7-8.

[0123] In FIG. 7A, the touch sensitive displays 110, 114 are in portrait display orientation and display, respectively, window 1 and second desktop D2. A gesture 700 is received by the gesture capture region 120, 124. Alternatively, the gesture 700 is received by a touch sensitive display 110, 114. The gesture may be any suitable gesture, including, without limitation, those gestures described above. By the gesture 700, the user indicated his or her command to move the window 1 from (origin) touch sensitive display 110 to (target) touch sensitive display 114.

[0124] Referring to FIG. 7B, in continuing response to gesture receipt the transition indicator 704 initiates movement lefr-to-right or right-to-left (depending on the orientation of the device 100 and touch sensitive display in question) typically from a point that appears to be behind the window 1 and typically at the same rate of movement as the tracked user gesture. Typically, the transition indicator 704 is not in the previously displayed image stack associated with the (origin) touch sensitive display 110. Stated another way, when the gesture is received, the transition indicator is not present in an active or inactive display position of either touch sensitive display 110 or 114. When and as the transition indicator 704 expands or moves, a seam 708 between the first and second touch sensitive displays 110, 114 and their respective displayed images is dimmed out completely to reveal the transition indicator background. Stated another way, the

transition indicator 704 moves outward from the seam 708 in a direction ultimately to cover, typically substantially completely, the touch sensitive display 114 and second desktop D2. As shown in FIGS. 7B and 7C, the transition indicator begins to cover the second desktop D2.

[0125] In FIG. 7D, the transition indicator 704 has covered fully the second desktop D2 in touch sensitive display 114 while the window 1 remains stationary in the touch sensitive display 110. Stated another way, window 1 is in the active display position for touch sensitive display 110 while the transition indicator 704 is in the active display position for touch sensitive display 114. Both the first and second desktops D1 and D2 are in inactive display positions for touch sensitive displays 110 and 114, respectively. In other configurations, the transition indicator covers only part of the target touch-sensitive display (114) before window movement is enabled or initiated.

[0126] When the transition indicator 704 has moved and occupied partially or fully the target touch sensitive display (in this example touch sensitive display 114) so that the prior display image (the prior display image is, in the example, the second desktop D2) is partly or completely dimmed out or covered by the transition indicator 704, the first window 1, as shown progressively by FIGS. 7E and 7F, moves over to occupy the target touch sensitive display 114. Until movement is triggered by proper placement of the transition indicator, the first window 1 remains as the displayed image for the source touch sensitive display (in this example touch sensitive display 110). Stated another way, the window 1 remains in the active display position for touch sensitive display 110 until the transition indicator has completed its movement to the target touch sensitive display 114. At that time, the window 1 slides over to cover the transition indicator 704, slowly uncovering the first desktop D1to occupy the active display position for touch sensitive display 114. When the window 1 has completed its movement to the target touch sensitive display 114, the first desktop D1 is in the active display position for touch sensitive display 110, and the second desktop D2 is in the inactive display position for touch sensitive display 114.

[0127] FIGS. 8A-E show the above steps for a device 100 in a landscape display orientation where the first window is being maximized to cover at least part of both the first and second touch sensitive displays 110 and 114. In FIG. 8A, the touch sensitive displays 110, 114 display, respectively, window 1 and second desktop D2. A gesture 700 is received by the gesture capture region 120, 124. Alternatively, the gesture 700 is received by a touch sensitive display 110, 114.

The gesture may be any suitable gesture, including, without limitation, those gestures described above. By the gesture 700, the user indicated his or her command to move the window 1 from touch sensitive display 110 to touch sensitive display 114.

[0128] Referring to FIG. 8B, in response to gesture receipt the transition indicator 704 initiates movement from a point that appears to be behind the window 1 up-to-down or down-to-up (depending on the orientation of the device 100 and touch sensitive display in question). When and as the transition indicator 704 expands or moves, a seam 708 between the first and second touch sensitive displays 110, 114 and their respective displayed images is dimmed out completely to reveal the transition indicator background. As shown in FIG. 8B, the transition indicator begins to cover the viewable area of the second desktop D2.

[0129] In FIG. 8C, the transition indicator 704 has covered partially or fully the second desktop D2 in touch sensitive display 114 while the window 1 remains stationary in the touch sensitive display 110. Stated another way, window 1 is in the active display position for touch sensitive display 110 while the transition indicator 704 is in the active display position for touch sensitive display 114. Both the first and second desktops D1 and D2 are in inactive display positions for touch sensitive displays 110 and 114, respectively.

[0130] When the transition indicator 704 has moved and occupied partly or fully the target touch sensitive display (in this example touch sensitive display 114) so that the prior display image (the prior display image is, in the example, the second desktop D2) is completely dimmed out or covered by the transition indicator 704, the first window 1, as shown progressively by FIGS. 8D and 8E, moves over to occupy the target touch sensitive display 114 as well as the origin touch sensitive display 110. Until that time, the first window 1 remains as the displayed image for the source touch sensitive display (in this example touch sensitive display 110). Stated another way, the window 1 remains in the active display position for touch sensitive display 110 until the transition indicator has completed its movement to the target touch sensitive display 114. At that time, the window 1 slides over to cover the transition indicator 704 to occupy the active display positions for both the origin and target touch sensitive displays 110 and 114.

[0131] In the various examples, middleware 520, particularly one or more of the Multi-Display Management (MDM) class 524, a Surface Cache class 528, a Window Management class 532, an Activity Management class 536, and an Application Management class 540, independently or collectively, detect receipt of a user gestures (step 900 of FIG. 9) and determine that the received

gesture commands movement of a displayed image, such as a window or desktop, to a target touch sensitive display. In response, the middleware 520 causes movement of the transition indicator 704 from the origin touch sensitive display to the target touch-sensitive display (step 1904). When the transition indicator 704 has covered a selected extent of the target touch sensitive display, the middleware 520 moves the displayed image to cover the target touch sensitive display (and to cover the transition indicator). Logic terminates in step 1912.

[0132] The exemplary systems and methods of this disclosure have been described in relation to communication devices. However, to avoid unnecessarily obscuring the present disclosure, the preceding description omits a number of known structures and devices. This omission is not to be construed as a limitation of the scopes of the claims. Specific details are set forth to provide an understanding of the present disclosure. It should however be appreciated that the present disclosure may be practiced in a variety of ways beyond the specific detail set forth herein.

[0133] Furthermore, while the exemplary aspects, embodiments, and/or configurations illustrated herein show the various components of the system collocated, certain components of the system can be located remotely, at distant portions of a distributed network, such as a LAN and/or the Internet, or within a dedicated system. Thus, it should be appreciated, that the components of the system can be combined in to one or more devices, such as a communication device, or collocated on a particular node of a distributed network, such as an analog and/or digital telecommunications network, a packet-switch network, or a circuit-switched network. It will be appreciated from the preceding description, and for reasons of computational efficiency, that the components of the system can be arranged at any location within a distributed network of components without affecting the operation of the system. For example, the various components can be located in a switch such as a PBX and media server, gateway, in one or more communications devices, at one or more users' premises, or some combination thereof. Similarly, one or more functional portions of the system could be distributed between a telecommunications device(s) and an associated computing device.

[0134] Furthermore, it should be appreciated that the various links connecting the elements can be wired or wireless links, or any combination thereof, or any other known or later developed element(s) that is capable of supplying and/or communicating data to and from the connected elements. These wired or wireless links can also be secure links and may be capable of

communicating encrypted information. Transmission media used as links, for example, can be any suitable carrier for electrical signals, including coaxial cables, copper wire and fiber optics, and may take the form of acoustic or light waves, such as those generated during radio-wave and infra-red data communications.

[0135] Also, while the flowcharts have been discussed and illustrated in relation to a particular sequence of events, it should be appreciated that changes, additions, and omissions to this sequence can occur without materially affecting the operation of the disclosed embodiments, configuration, and aspects.

[0136] A number of variations and modifications of the disclosure can be used. It would be possible to provide for some features of the disclosure without providing others.

[0137] For example in one alternative embodiment, movement of displayed images other than windows and desktops are telegraphed by prior movement of the transition indicator.

[0138] In other embodiments, the transition indicator previews window maximization from only one touch-sensitive display to covering both touch-sensitive displays.

[0139] In another alternative embodiment, the transition indicator, during transitions, covers the entire touch sensitive display when the device 100 is closed and only the primary screen is active and/or covers both touch sensitive displays when the device 100 is open. The latter case can occur, for example, when a window is maximized or opened and covers both touch-sensitive displays or when the transition affects both the primary and secondary screens. When a window is maximized onto a single touch sensitive display of a closed device or over both touch sensitive displays of an open device, the transition indicator can move from an edge over the active touch sensitive display(s).

[0140] In other embodiments, the present disclosure applies to other displayed image transitions besides window movement. In such transitions, at least a portion the touch sensitive display changes a displayed image. The change of the displayed image is indicated by covering the at least a portion of the display with a transition indicator after the previously displayed image is removed and before the new displayed image is introduced.

[0141] In yet another embodiment, the systems and methods of this disclosure can be implemented in conjunction with a special purpose computer, a programmed microprocessor or microcontroller and peripheral integrated circuit element(s), an ASIC or other integrated circuit, a digital signal processor, a hard-wired electronic or logic circuit such as discrete element circuit,

a programmable logic device or gate array such as PLD, PLA, FPGA, PAL, special purpose computer, any comparable means, or the like. In general, any device(s) or means capable of implementing the methodology illustrated herein can be used to implement the various aspects of this disclosure. Exemplary hardware that can be used for the disclosed embodiments, configurations and aspects includes computers, handheld devices, telephones (e.g., cellular, Internet enabled, digital, analog, hybrids, and others), and other hardware known in the art. Some of these devices include processors (e.g., a single or multiple microprocessors), memory, nonvolatile storage, input devices, and output devices. Furthermore, alternative software implementations including, but not limited to, distributed processing or component/object distributed processing, parallel processing, or virtual machine processing can also be constructed to implement the methods described herein.

[0142] In yet another embodiment, the disclosed methods may be readily implemented in conjunction with software using object or object-oriented software development environments that provide portable source code that can be used on a variety of computer or workstation platforms. Alternatively, the disclosed system may be implemented partially or fully in hardware using standard logic circuits or VLSI design. Whether software or hardware is used to implement the systems in accordance with this disclosure is dependent on the speed and/or efficiency requirements of the system, the particular function, and the particular software or hardware systems or microprocessor or microcomputer systems being utilized.

[0143] In yet another embodiment, the disclosed methods may be partially implemented in software that can be stored on a storage medium, executed on programmed general-purpose computer with the cooperation of a controller and memory, a special purpose computer, a microprocessor, or the like. In these instances, the systems and methods of this disclosure can be implemented as program embedded on personal computer such as an applet, JAVA® or CGI script, as a resource residing on a server or computer workstation, as a routine embedded in a dedicated measurement system, system component, or the like. The system can also be implemented by physically incorporating the system and/or method into a software and/or hardware system.

[0144] Although the present disclosure describes components and functions implemented in the aspects, embodiments, and/or configurations with reference to particular standards and protocols, the aspects, embodiments, and/or configurations are not limited to such standards and

protocols. Other similar standards and protocols not mentioned herein are in existence and are considered to be included in the present disclosure. Moreover, the standards and protocols mentioned herein and other similar standards and protocols not mentioned herein are periodically superseded by faster or more effective equivalents having essentially the same functions. Such replacement standards and protocols having the same functions are considered equivalents included in the present disclosure.

[0145] The present disclosure, in various aspects, embodiments, and/or configurations, includes components, methods, processes, systems and/or apparatus substantially as depicted and described herein, including various aspects, embodiments, configurations embodiments, subcombinations, and/or subsets thereof. Those of skill in the art will understand how to make and use the disclosed aspects, embodiments, and/or configurations afrer understanding the present disclosure. The present disclosure, in various aspects, embodiments, and/or configurations, includes providing devices and processes in the absence of items not depicted and/or described herein or in various aspects, embodiments, and/or configurations hereof, including in the absence of such items as may have been used in previous devices or processes, e.g., for improving performance, achieving ease and\or reducing cost of implementation.

[0146] The foregoing discussion has been presented for purposes of illustration and description. The foregoing is not intended to limit the disclosure to the form or forms disclosed herein. In the foregoing Detailed Description for example, various features of the disclosure are grouped together in one or more aspects, embodiments, and/or configurations for the purpose of streamlining the disclosure. The features of the aspects, embodiments, and/or configurations of the disclosure may be combined in alternate aspects, embodiments, and/or configurations other than those discussed above. This method of disclosure is not to be interpreted as reflecting an intention that the claims require more features than are expressly recited in each claim. Rather, as the following claims reflect, inventive aspects lie in less than all features of a single foregoing disclosed aspect, embodiment, and/or configuration. Thus, the following claims are hereby incorporated into this Detailed Description, with each claim standing on its own as a separate preferred embodiment of the disclosure.

[0147] Moreover, though the description has included description of one or more aspects, embodiments, and/or configurations and certain variations and modifications, other variations, combinations, and modifications are within the scope of the disclosure, e.g., as may be within the

skill and knowledge of those in the art, afrer understanding the present disclosure. It is intended to obtain rights which include alternative aspects, embodiments, and/or configurations to the extent permitted, including alternate, interchangeable and/or equivalent structures, functions, ranges or steps to those claimed, whether or not such alternate, interchangeable and/or equivalent structures, functions, ranges or steps are disclosed herein, and without intending to publicly dedicate any patentable subject matter.

What is claimed is:

1. A method, comprising:

receiving, by at least one of a gesture capture region and a touch sensitive display, a gesture, the gesture indicating that a displayed image is to be moved from a first touch sensitive display to a second touch sensitive display; and

in response and prior to movement of the displayed image to the second touch sensitive display, moving, by a microprocessor, a transition indicator from the first touch sensitive display to the second touch sensitive display to a selected position to be occupied by the displayed image; and

thereafrer moving, by the microprocessor, the displayed image from the first touch sensitive display to the second touch sensitive display to the selected position.

2. The method of claim 1, wherein the displayed image is a desktop and wherein the gesture is received by the touch sensitive display.

3. The method of claim 1, wherein the displayed image is a window, wherein the gesture is received by the gesture capture region, and wherein the gesture capture region is unable to display any displayed image.

4. The method of claim 1, wherein the transition indicator moves along a path of travel of the displayed image to the selected position to preview movement of the displayed image and wherein a size and shape of the transition indicator is substantially the same as the displayed image.

5. The method of claim 1, wherein the transition indicator is unable to receive or provide dynamic user input or output, respectively, wherein the transition indicator has an appearance different from the displayed image, and wherein the transition indicator covers only part of the second touch-sensitive display before the displayed image is moved to cover fully the second touch-sensitive display.

6. The method of claim 1, wherein, when the gesture is received, the transition indicator is absent from a displayed image stack associated with the first and second touch sensitive displays, wherein the displayed image and transition indicator are simultaneously in active display positions on the first and second touch sensitive displays, respectively, prior to initiation of movement of the displayed image and wherein the transition indicator comprises a user configured color, pattern, design, and/or photograph.

7. The method of claim 1, wherein the transition indicator is a graphical affordance, wherein the window is controlled by a multi-screen application, and wherein the transition indicator, during movement from the first touch sensitive display to the second touch sensitive display, is substantially non-responsive to a user command or request.

8. A non-transient computer readable medium comprising microprocessor executable instructions operable to perform at least the following steps:

receiving, by at least one of a gesture capture region and a touch sensitive display, a gesture, the gesture indicating that a displayed image is to be moved from a first touch sensitive display to a second touch sensitive display; and

in response and prior to movement of the displayed image to the second touch sensitive display, moving a transition indicator from the first touch sensitive display to the second touch sensitive display to a selected position to be occupied by the displayed image; and

thereafter moving the displayed image from the first touch sensitive display to the second touch sensitive display to the selected position.

9. The medium of claim 8, wherein the displayed image is a desktop and wherein the gesture is received by the touch sensitive display.

10. The medium of claim 8, wherein the displayed image is a window, wherein the gesture is received by the gesture capture region, and wherein the gesture capture region is unable to display any displayed image.

11. The medium of claim 8 wherein the transition indicator moves along a path of travel of the displayed image to the selected position to preview movement of the displayed image and wherein a size and shape of the transition indicator is substantially the same as the displayed image.

12. The medium of claim 8 wherein the transition indicator is unable to receive or provide dynamic user input or output, respectively, wherein the transition indicator has an appearance different from the displayed image, and wherein the transition indicator covers only part of the second touch-sensitive display before the displayed image is moved to cover fully the second touch-sensitive display.

13. The medium of claim 8, wherein, when the gesture is received, the transition indicator is absent from a displayed image stack associated with the first and second touch sensitive displays, wherein the displayed image and transition indicator are simultaneously in

active display positions on the first and second touch sensitive displays, respectively, prior to initiation of movement of the displayed image and wherein the transition indicator comprises a user configured color, pattern, design, and/or photograph.

14. The medium of claim 8, wherein the transition indicator is a graphical affordance, wherein the window is controlled by a multi-screen application, and wherein the transition indicator, during movement from the first touch sensitive display to the second touch sensitive display, is substantially non-responsive to a user command or request.

15. A dual display communication device, comprising:

a gesture capture region to receive a gesture;

a first touch sensitive display to receive a gesture and display displayed images, wherein the displayed images are at least one of a desktop and window of an application;

a second touch sensitive display to receive a gesture and display displayed images; and

middleware operable to perform at least one of the following operations:

receive a gesture, the gesture indicating that a displayed image is to be expanded from the first touch sensitive display to cover at least most of the second touch sensitive display;

in response and prior to expansion of the displayed image to the second touch sensitive display, expand a transition indicator to cover at least most of the second touch sensitive display; and

thereafrer expand the displayed image to the second touch sensitive display.

16. The device of claim 15, wherein the displayed image is a desktop and wherein the gesture is received by the touch sensitive display.

17. The device of claim 15, wherein the displayed image is a minimized window, wherein the gesture is received by the gesture capture region, wherein the gesture capture region is unable to display any displayed image, and wherein the window is maximized to cover at least most of the first and second touch-sensitive displays.

18. The device of claim 15, wherein the transition indicator moves along a path of travel of the displayed image to the selected position to preview expansion of the displayed image and wherein a size and shape of the transition indicator is substantially the same as the displayed image.

19. The device of claim 15, wherein the transition indicator is unable to receive or provide dynamic user input or output, respectively, and wherein the transition indicator has an appearance different from the displayed image.

20. The device of claim 15, wherein, when the gesture is received, the transition indicator is absent from a displayed image stack associated with the first and second touch sensitive displays, wherein the displayed image and transition indicator are simultaneously in active display positions on the first and second touch sensitive displays, respectively, prior to initiation of expansion of the displayed image and wherein the transition indicator comprises a user configured color, pattern, design, and/or photograph.

21. The device of claim 15, wherein the transition indicator is a graphical affordance, wherein the window is controlled by a multi-screen application, and wherein the transition indicator, during expansion from the first touch sensitive display to the second touch sensitive display, is substantially non-responsive to a user command or request.

ABSTRACT

[0148] A dual display communication device includes a gesture capture region to receive a gesture, a first touch sensitive display to receive a gesture and display displayed images (such as a desktop or window of an application), and a second touch sensitive display to receive a gesture and display displayed images. Middleware receives a gesture, the gesture indicating that a displayed image is to be moved from the first touch sensitive display to the second touch sensitive display, such as to maximize a window to cover portions of both displays simultaneously; in response and prior to movement of the displayed image to the second touch sensitive display, moves a transition indicator from the first touch sensitive display to the second touch sensitive display to a selected position to be occupied by the displayed image; and thereafter moves the displayed image from the first touch sensitive display to the second touch sensitive display to the selected position.

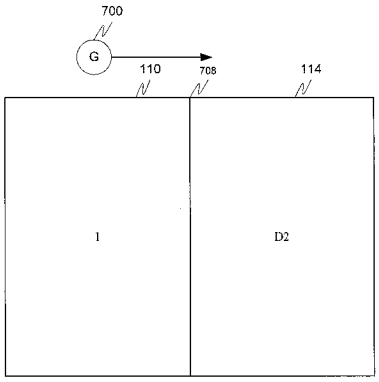


Fig. 7A

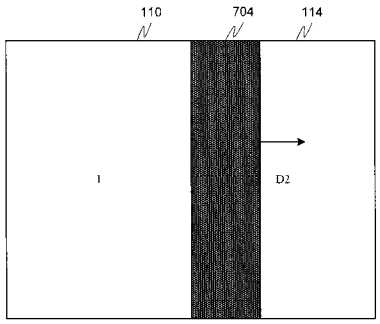


Fig. 7B

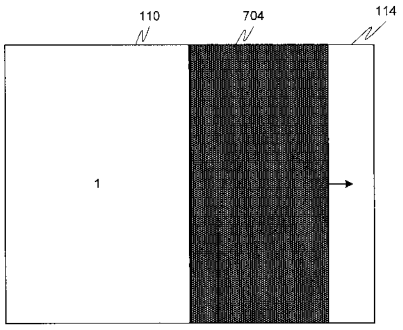


Fig. 7C

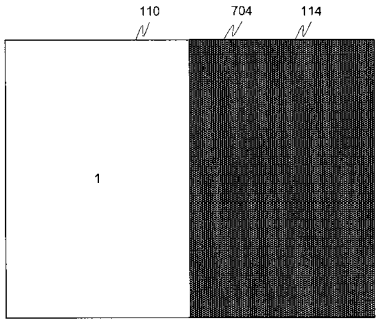


Fig. 7D

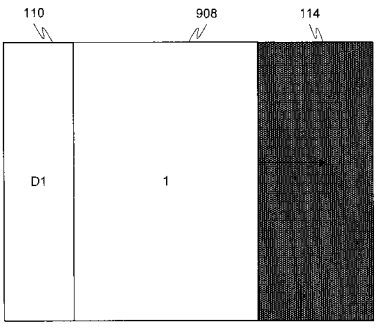


Fig. 7E

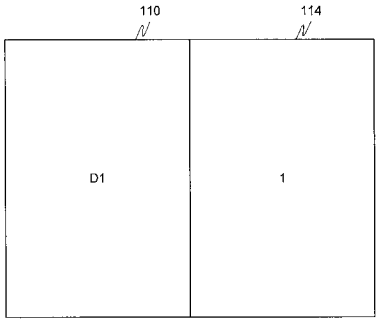


Fig. 7F

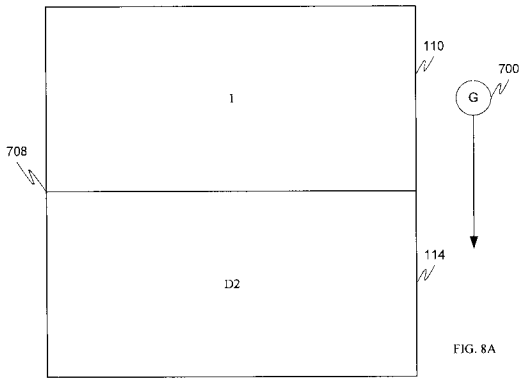


FIG. 8A

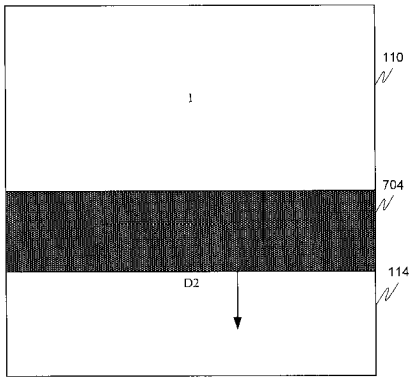


FIG. 8B

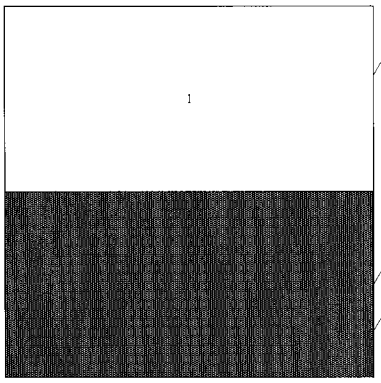


FIG. 8C

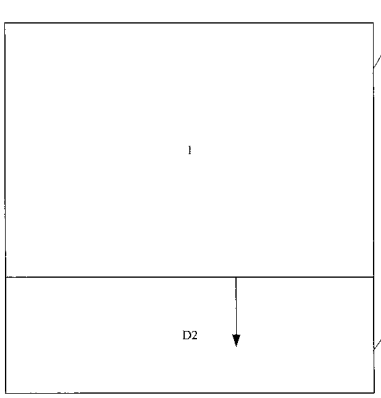


FIG. 8D

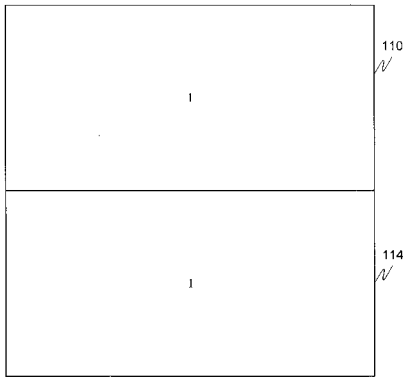


FIG. 8E

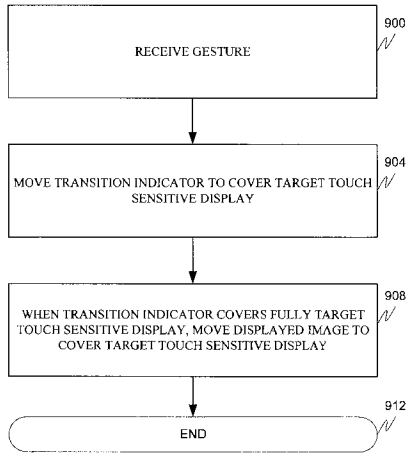


FIG. 9

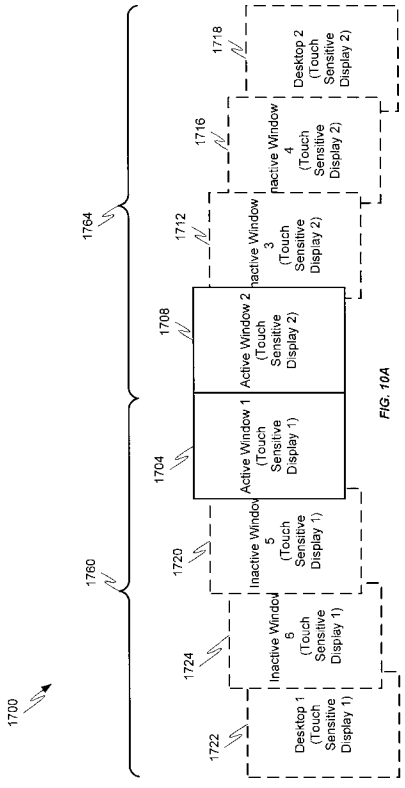


FIG. 10A

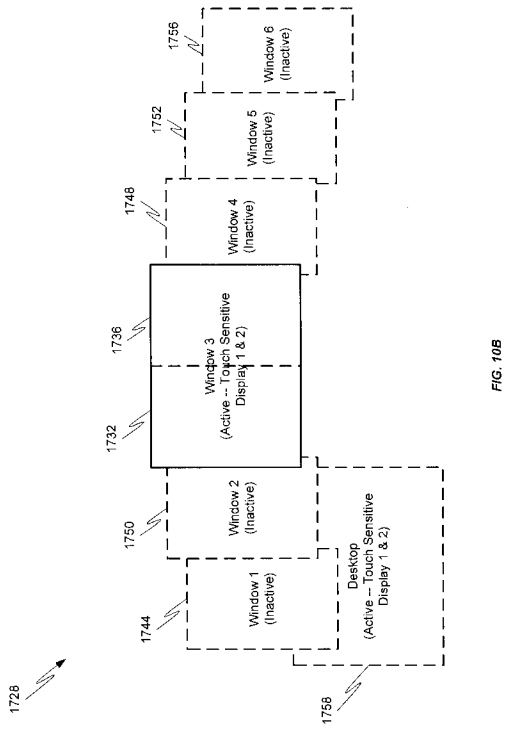


FIG. 10B

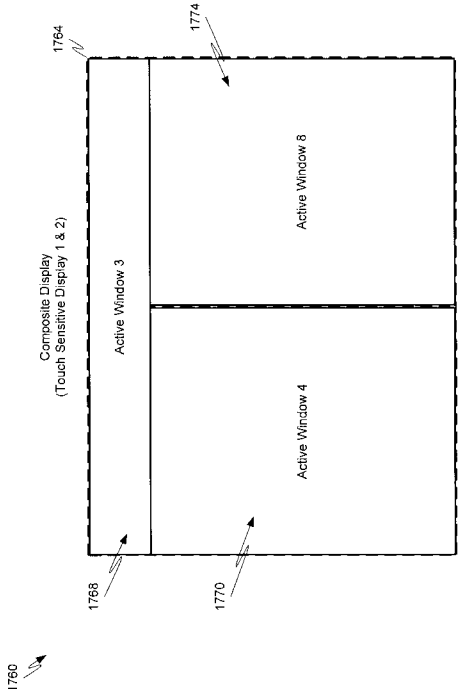


FIG. 10C

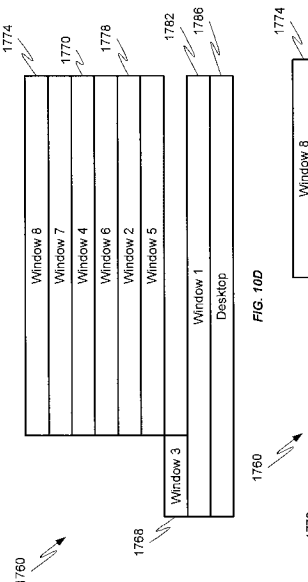


FIG. 10D

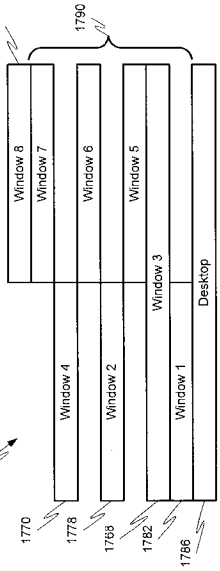


FIG. 10E

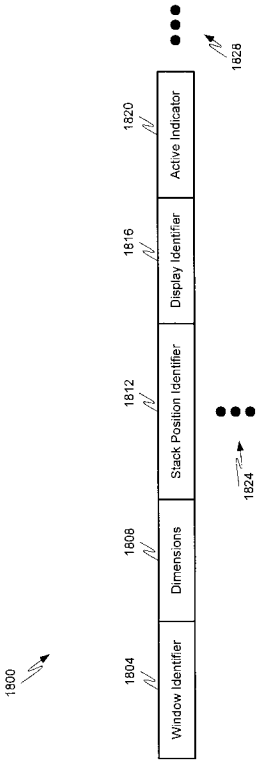


FIG. 11

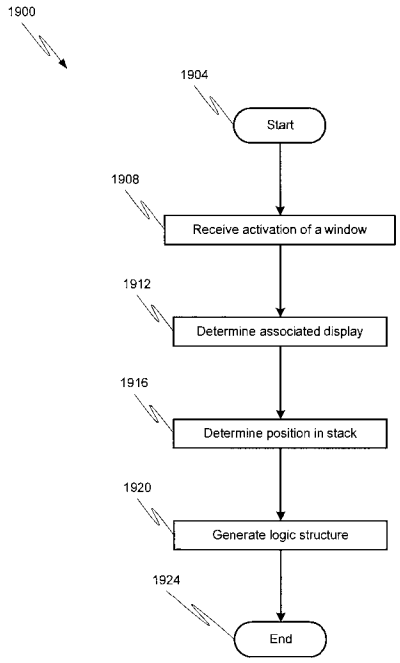

1900
1904
Start
1908
Receive activation of a window
1912
Determine associated display
1916
Determine position in stack
1920
Generate logic structure
1924
End

FIG. 12

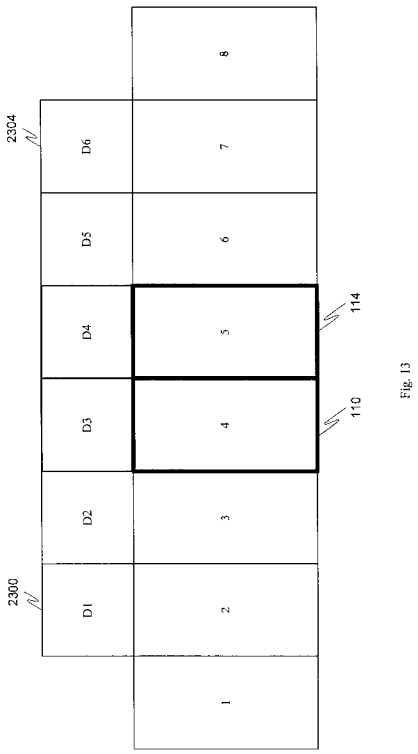

2304
D6
D5
D4
D3
D2
D1
2300
8
7
6
5
4
3
2
1
114
110

Fig. 13

PTO/SB/14 (11-08)
Approved for use through 01/31/2014. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

Application Data Sheet 37 CFR 1.76	Attorney Docket Number	6583
	Application Number	
Title of Invention	MOBILE DEVICE	

The application data sheet is part of the provisional or nonprovisional application for which it is being submitted. The following form contains the bibliographic data arranged in a format specified by the United States Patent and Trademark Office as outlined in 37 CFR 1.76.
This document may be completed electronically and submitted to the Office in electronic format using the Electronic Filing System (EFS) or the document may be printed and included in a paper filed application.

Secrecy Order 37 CFR 5.2

☐ Portions or all of the application associated with this Application Data Sheet may fall under a Secrecy Order pursuant to 37 CFR 5.2 (Paper filers only. Applications that fall under Secrecy Order may not be filed electronically.)

Applicant Information:

Applicant 1 Remove

Applicant Authority ◉ Inventor ○ Legal Representative under 35 U.S.C. 117 ○ Party of Interest under 35 U.S.C. 118

Prefix	Given Name	Middle Name	Family Name	Suffix
	Denise		Burton	

Residence Information (Select One) ◉ US Residency ○ Non US Residency ○ Active US Military Service

City	Austin	State/Province	TX	Country of Residence[i]	US

Citizenship under 37 CFR 1.41(b)[i] US

Mailing Address of Applicant:

Address 1	14816 Bear Creek Pass		
Address 2			
City	Austin	State/Province	TX
Postal Code	78737	Country[i]	US

Applicant 2 Remove

Applicant Authority ◉ Inventor ○ Legal Representative under 35 U.S.C. 117 ○ Party of Interest under 35 U.S.C. 118

Prefix	Given Name	Middle Name	Family Name	Suffix
	Ron		Cassar	

Residence Information (Select One) ○ US Residency ◉ Non US Residency ○ Active US Military Service

City	Burlington	Country Of Residence[i]	CA

Citizenship under 37 CFR 1.41(b)[i] CA

Mailing Address of Applicant:

Address 1	165 Nicholson Crt		
Address 2			
City	Burlington	State/Province	ON
Postal Code	L7N 3N5	Country[i]	CA

Applicant 3 Remove

Applicant Authority ◉ Inventor ○ Legal Representative under 35 U.S.C. 117 ○ Party of Interest under 35 U.S.C. 118

Prefix	Given Name	Middle Name	Family Name	Suffix
	Tong		Chen	

Residence Information (Select One) ○ US Residency ◉ Non US Residency ○ Active US Military Service

City	Markham	Country Of Residence[i]	CA

Application Data Sheet 37 CFR 1.76	Attorney Docket Number	6583
	Application Number	
Title of Invention	MOBILE DEVICE	

Citizenship under 37 CFR 1.41(b) i	CA				
Mailing Address of Applicant:					
Address 1	19 Campden Place				
Address 2					
City	Markham	**State/Province**	ON		
Postal Code	L6C 2J4	**Country**i	CA		

Applicant 4 Remove

Applicant Authority ◉ Inventor	○ Legal Representative under 35 U.S.C. 117	○ Party of Interest under 35 U.S.C. 118	
Prefix	**Given Name**	**Middle Name**	**Family Name** / **Suffix**
	Misty		Cripps

Residence Information (Select One) ◉ US Residency ○ Non US Residency ○ Active US Military Service

City	Austin	**State/Province**	TX	**Country of Residence** i	US
Citizenship under 37 CFR 1.41(b) i	US				
Mailing Address of Applicant:					
Address 1	2610 Northeast Drive				
Address 2					
City	Austin	**State/Province**	TX		
Postal Code	78723	**Country**i	US		

Applicant 5 Remove

Applicant Authority ◉ Inventor	○ Legal Representative under 35 U.S.C. 117	○ Party of Interest under 35 U.S.C. 118	
Prefix	**Given Name**	**Middle Name**	**Family Name** / **Suffix**
	Robert		Csiki

Residence Information (Select One) ○ US Residency ◉ Non US Residency ○ Active US Military Service

City	Markham	**Country Of Residence**i	CA
Citizenship under 37 CFR 1.41(b) i	CA		
Mailing Address of Applicant:			
Address 1	131 Major Button's Dr		
Address 2			
City	Markham	**State/Province**	ON
Postal Code	L3P 3X7	**Country**i	CA

Applicant 6 Remove

Applicant Authority ◉ Inventor	○ Legal Representative under 35 U.S.C. 117	○ Party of Interest under 35 U.S.C. 118	
Prefix	**Given Name**	**Middle Name**	**Family Name** / **Suffix**
	Alexander		de Paz

Residence Information (Select One) ○ US Residency ◉ Non US Residency ○ Active US Military Service

City	Burlington	**Country Of Residence**i	CA
Citizenship under 37 CFR 1.41(b) i	CA		

Application Data Sheet 37 CFR 1.76	Attorney Docket Number	6583
	Application Number	
Title of Invention	MOBILE DEVICE	

Mailing Address of Applicant:			
Address 1	5444 Valleyhigh Drive		
Address 2			
City	Burlington	State/Province	ON
Postal Code	L7L 6Z6	Country[i]	CA

Remove

Applicant 7

Applicant Authority	(•) Inventor	( ) Legal Representative under 35 U.S.C. 117	( ) Party of Interest under 35 U.S.C. 118

Prefix	Given Name	Middle Name	Family Name	Suffix
	Brett	Bernard	Faulk	

Residence Information (Select One)	(•) US Residency	( ) Non US Residency	( ) Active US Military Service

City	Cypress	State/Province	TX	Country of Residence[i]	US
Citizenship under 37 CFR 1.41(b)[i]	US				

Mailing Address of Applicant:			
Address 1	14226 Blair Ridge Drive		
Address 2			
City	Cypress	State/Province	TX
Postal Code	77429	Country[i]	US

Remove

Applicant 8

Applicant Authority	(•) Inventor	( ) Legal Representative under 35 U.S.C. 117	( ) Party of Interest under 35 U.S.C. 118

Prefix	Given Name	Middle Name	Family Name	Suffix
	Jared	L.	Ficklin	

Residence Information (Select One)	(•) US Residency	( ) Non US Residency	( ) Active US Military Service

City	Austin	State/Province	TX	Country of Residence[i]	US
Citizenship under 37 CFR 1.41(b)[i]	US				

Mailing Address of Applicant:			
Address 1	4213 Parkwood Road		
Address 2			
City	Austin	State/Province	TX
Postal Code	78722	Country[i]	US

Remove

Applicant 9

Applicant Authority	(•) Inventor	( ) Legal Representative under 35 U.S.C. 117	( ) Party of Interest under 35 U.S.C. 118

Prefix	Given Name	Middle Name	Family Name	Suffix
	Eric		Freedman	

Residence Information (Select One)	(•) US Residency	( ) Non US Residency	( ) Active US Military Service

City	Austin	State/Province	TX	Country of Residence[i]	US
Citizenship under 37 CFR 1.41(b)[i]	US				

PTO/SB/14 (11-08)
Approved for use through 01/31/2014. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

Application Data Sheet 37 CFR 1.76	Attorney Docket Number	6583
	Application Number	
Title of Invention	MOBILE DEVICE	

Mailing Address of Applicant:			
Address 1	4417 RidgeOak Drive		
Address 2			
City	Austin	State/Province	TX
Postal Code	78731	Country[i]	US

Applicant 10 — Remove

Applicant Authority: (•) Inventor ( ) Legal Representative under 35 U.S.C. 117 ( ) Party of Interest under 35 U.S.C. 118

Prefix	Given Name	Middle Name	Family Name	Suffix
	Martin		Gimpl	

Residence Information (Select One): ( ) US Residency (•) Non US Residency ( ) Active US Military Service

City	Helsinki	Country Of Residence[i]	FI
Citizenship under 37 CFR 1.41(b)[i]	FI		

Mailing Address of Applicant:			
Address 1	Jussaarenkuja 5 C 32		
Address 2			
City	Helsinki	State/Province	
Postal Code	00840	Country[i]	FI

Applicant 11 — Remove

Applicant Authority: (•) Inventor ( ) Legal Representative under 35 U.S.C. 117 ( ) Party of Interest under 35 U.S.C. 118

Prefix	Given Name	Middle Name	Family Name	Suffix
	Volodimir	Felixovich	Lemberg	

Residence Information (Select One): ( ) US Residency (•) Non US Residency ( ) Active US Military Service

City	Etobicoke	Country Of Residence[i]	CA
Citizenship under 37 CFR 1.41(b)[i]	CA		

Mailing Address of Applicant:			
Address 1	631 Evans Ave.		
Address 2			
City	Etobicoke	State/Province	ON
Postal Code	M8W 2W4	Country[i]	CA

Applicant 12 — Remove

Applicant Authority: (•) Inventor ( ) Legal Representative under 35 U.S.C. 117 ( ) Party of Interest under 35 U.S.C. 118

Prefix	Given Name	Middle Name	Family Name	Suffix
	Maxim		Marintchenko	

Residence Information (Select One): ( ) US Residency (•) Non US Residency ( ) Active US Military Service

City	Vaughan	Country Of Residence[i]	CA
Citizenship under 37 CFR 1.41(b)[i]	CA		

Application Data Sheet 37 CFR 1.76	Attorney Docket Number	6583
	Application Number	
Title of Invention	MOBILE DEVICE	

Mailing Address of Applicant:			
Address 1	35 Chardonnay Drive		
Address 2			
City	Vaughan	State/Province	ON
Postal Code	L4J 8R8	Country[i]	CA

Remove

Applicant 13

Applicant Authority (•) Inventor ( ) Legal Representative under 35 U.S.C. 117 ( ) Party of Interest under 35 U.S.C. 118

Prefix	Given Name	Middle Name	Family Name	Suffix
	Eduardo	Diego Torres	Milano	

Residence Information (Select One) ( ) US Residency (•) Non US Residency ( ) Active US Military Service

City	Burlington	Country Of Residence[i]	CA
Citizenship under 37 CFR 1.41(b)[i]	AR		

Mailing Address of Applicant:			
Address 1	629 Marley Road		
Address 2			
City	Burlington	State/Province	ON
Postal Code	L7T 3R8	Country[i]	CA

Remove

Applicant 14

Applicant Authority (•) Inventor ( ) Legal Representative under 35 U.S.C. 117 ( ) Party of Interest under 35 U.S.C. 118

Prefix	Given Name	Middle Name	Family Name	Suffix
	Paul	Edward	Reeves	

Residence Information (Select One) ( ) US Residency (•) Non US Residency ( ) Active US Military Service

City	Oakville	Country Of Residence[i]	CA
Citizenship under 37 CFR 1.41(b)[i]	CA		

Mailing Address of Applicant:			
Address 1	1215 Riverbank Way		
Address 2			
City	Oakville	State/Province	ON
Postal Code	L6H 6X4	Country[i]	CA

Remove

Applicant 15

Applicant Authority (•) Inventor ( ) Legal Representative under 35 U.S.C. 117 ( ) Party of Interest under 35 U.S.C. 118

Prefix	Given Name	Middle Name	Family Name	Suffix
	Rodney	Wayne	Schrock	

Residence Information (Select One) (•) US Residency ( ) Non US Residency ( ) Active US Military Service

City	San Diego	State/Province	CA	Country of Residence[i]	US
Citizenship under 37 CFR 1.41(b)[i]	US				

Application Data Sheet 37 CFR 1.76	Attorney Docket Number	6583
	Application Number	
Title of Invention	MOBILE DEVICE	

Mailing Address of Applicant:			
Address 1	13921 Rancho Dorado Bend		
Address 2			
City	San Diego	State/Province	CA
Postal Code	92130	Country[i]	US

Applicant 16 — Remove

Applicant Authority	◉ Inventor	○ Legal Representative under 35 U.S.C. 117	○ Party of Interest under 35 U.S.C. 118

Prefix	Given Name	Middle Name	Family Name	Suffix
	Mohammed		Selim	

Residence Information (Select One) ○ US Residency ◉ Non US Residency ○ Active US Military Service

City	Oakville	Country Of Residence[i]	CA
Citizenship under 37 CFR 1.41(b) [i]	CA		

Mailing Address of Applicant:			
Address 1	2488 Oakhaven Drive,		
Address 2			
City	Oakville	State/Province	ON
Postal Code	L6M4B9	Country[i]	CA

Applicant 17 — Remove

Applicant Authority	◉ Inventor	○ Legal Representative under 35 U.S.C. 117	○ Party of Interest under 35 U.S.C. 118

Prefix	Given Name	Middle Name	Family Name	Suffix
	Sanjiv		Sirpal	

Residence Information (Select One) ○ US Residency ◉ Non US Residency ○ Active US Military Service

City	Oakville	Country Of Residence[i]	CA
Citizenship under 37 CFR 1.41(b) [i]	CA		

Mailing Address of Applicant:			
Address 1	3264 Stocksbridge Avenue		
Address 2			
City	Oakville	State/Province	ON
Postal Code		Country[i]	CA

Applicant 18 — Remove

Applicant Authority	◉ Inventor	○ Legal Representative under 35 U.S.C. 117	○ Party of Interest under 35 U.S.C. 118

Prefix	Given Name	Middle Name	Family Name	Suffix
	Nikhil		Swaminathan	

Residence Information (Select One) ○ US Residency ◉ Non US Residency ○ Active US Military Service

City	Toronto	Country Of Residence[i]	CA
Citizenship under 37 CFR 1.41(b) [i]	CA		

Application Data Sheet 37 CFR 1.76	Attorney Docket Number	6583
	Application Number	
Title of Invention	MOBILE DEVICE	

Mailing Address of Applicant:			
Address 1	3210-65 Bremner Blvd		
Address 2			
City	Toronto	State/Province	ON
Postal Code	M5E2A1	Country[i]	CA

Applicant 19 — Remove

Applicant Authority	(•) Inventor	( ) Legal Representative under 35 U.S.C. 117	( ) Party of Interest under 35 U.S.C. 118

Prefix	Given Name	Middle Name	Family Name	Suffix
	John	Steven	Visosky	

Residence Information (Select One)	( ) US Residency	(•) Non US Residency	( ) Active US Military Service

City	Gormley	Country Of Residence[i]	CA
Citizenship under 37 CFR 1.41(b) [i]	CA		

Mailing Address of Applicant:			
Address 1	11 Lady Diana Court		
Address 2			
City	Gormley	State/Province	ON
Postal Code	L0H 1G0	Country[i]	CA

Applicant 20 — Remove

Applicant Authority	(•) Inventor	( ) Legal Representative under 35 U.S.C. 117	( ) Party of Interest under 35 U.S.C. 118

Prefix	Given Name	Middle Name	Family Name	Suffix
	Suxiang		Wei	

Residence Information (Select One)	( ) US Residency	(•) Non US Residency	( ) Active US Military Service

City	Mississauga	Country Of Residence[i]	CA
Citizenship under 37 CFR 1.41(b) [i]	CN		

Mailing Address of Applicant:			
Address 1	3855 Skyview St.		
Address 2			
City	Mississauga	State/Province	ON
Postal Code	L5M 8A1	Country[i]	CA

Applicant 21 — Remove

Applicant Authority	(•) Inventor	( ) Legal Representative under 35 U.S.C. 117	( ) Party of Interest under 35 U.S.C. 118

Prefix	Given Name	Middle Name	Family Name	Suffix
	Gregg		Wygonik	

Residence Information (Select One)	(•) US Residency	( ) Non US Residency	( ) Active US Military Service

City	Austin	State/Province	TX	Country of Residence [i]	US
Citizenship under 37 CFR 1.41(b) [i]	US				

Application Data Sheet 37 CFR 1.76	Attorney Docket Number	6583
	Application Number	
Title of Invention	MOBILE DEVICE	

Mailing Address of Applicant:			
Address 1	101 West 6th Street		
Address 2	2nd Floor		
City	Austin	State/Province	TX
Postal Code	78701	Country[i]	US

Applicant 22 — Remove

Applicant Authority: (•) Inventor ( ) Legal Representative under 35 U.S.C. 117 ( ) Party of Interest under 35 U.S.C. 118

Prefix	Given Name	Middle Name	Family Name	Suffix
	Miroslaw		Zaremski	

Residence Information (Select One): ( ) US Residency (•) Non US Residency ( ) Active US Military Service

City	Richmond Hill	Country Of Residence[i]	CA
Citizenship under 37 CFR 1.41(b)[i]	CA		

Mailing Address of Applicant:			
Address 1	40 Alyssum Court		
Address 2			
City	Richmond Hill	State/Province	ON
Postal Code	L4E 4M7	Country[i]	CA

All Inventors Must Be Listed - Additional Inventor Information blocks may be generated within this form by selecting the **Add** button. Add

Correspondence Information:

Enter either Customer Number or complete the Correspondence Information section below.
For further information see 37 CFR 1.33(a).

☐ **An Address is being provided for the correspondence Information of this application.**

Customer Number	22442		
Email Address	srlaw@sheridanross.com	Add Email	Remove Email

Application Information:

Title of the Invention	MOBILE DEVICE		
Attorney Docket Number	6583	Small Entity Status Claimed	☐
Application Type	Provisional		
Subject Matter	Utility		
Suggested Class (if any)		Sub Class (if any)	
Suggested Technology Center (if any)			
Total Number of Drawing Sheets (if any)	157	Suggested Figure for Publication (if any)	

PTO/SB/14 (11-08)
Approved for use through 01/31/2014. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

Application Data Sheet 37 CFR 1.76	Attorney Docket Number	6583
	Application Number	
Title of Invention	MOBILE DEVICE	

Publication Information:

☐	Request Early Publication (Fee required at time of Request 37 CFR 1.219)
☐	**Request Not to Publish.** I hereby request that the attached application not be published under 35 U.S.C. 122(b) and certify that the invention disclosed in the attached application **has not and will not** be the subject of an application filed in another country, or under a multilateral international agreement, that requires publication at eighteen months after filing.

Representative Information:

Representative information should be provided for all practitioners having a power of attorney in the application. Providing this information in the Application Data Sheet does not constitute a power of attorney in the application (see 37 CFR 1.32).
Enter either Customer Number or complete the Representative Name section below. If both sections are completed the Customer Number will be used for the Representative Information during processing.

Please Select One:	◉ Customer Number	○ US Patent Practitioner	○ Limited Recognition (37 CFR 11.9)
Customer Number	22442		

Domestic Benefit/National Stage Information:

This section allows for the applicant to either claim benefit under 35 U.S.C. 119(e), 120, 121, or 365(c) or indicate National Stage entry from a PCT application. Providing this information in the application data sheet constitutes the specific reference required by 35 U.S.C. 119(e) or 120, and 37 CFR 1.78(a)(2) or CFR 1.78(a)(4), and need not otherwise be made part of the specification.

Prior Application Status			Remove
Application Number	Continuity Type	Prior Application Number	Filing Date (YYYY-MM-DD)

Additional Domestic Benefit/National Stage Data may be generated within this form by selecting the **Add** button. Add

Foreign Priority Information:

This section allows for the applicant to claim benefit of foreign priority and to identify any prior foreign application for which priority is not claimed. Providing this information in the application data sheet constitutes the claim for priority as required by 35 U.S.C. 119(b) and 37 CFR 1.55(a).

Remove

Application Number	Country [i]	Parent Filing Date (YYYY-MM-DD)	Priority Claimed
			○ Yes ◉ No

Additional Foreign Priority Data may be generated within this form by selecting the **Add** button. Add

Assignee Information:

Providing this information in the application data sheet does not substitute for compliance with any requirement of part 3 of Title 37 of the CFR to have an assignment recorded in the Office.

Assignee 1 Remove

Application Data Sheet 37 CFR 1.76	Attorney Docket Number	6583
	Application Number	
Title of Invention	MOBILE DEVICE	

If the Assignee is an Organization check here. ☒			
Organization Name	IMERJ LTD.		
Mailing Address Information:			
Address 1	305 Interlocken Parkway		
Address 2			
City	Broomfield	**State/Province**	CO
Country [i]	US	Postal Code	80021
Phone Number		Fax Number	
Email Address			
Additional Assignee Data may be generated within this form by selecting the **Add** button.		Add	

Signature:

A signature of the applicant or representative is required in accordance with 37 CFR 1.33 and 10.18. Please see 37 CFR 1.4(d) for the form of the signature.					
Signature	/Tadd F. Wilson/			Date (YYYY-MM-DD)	2011-09-27
First Name	Tadd F.	Last Name	Wilson	Registration Number	54544

This collection of information is required by 37 CFR 1.76. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 23 minutes to complete, including gathering, preparing, and submitting the completed application data sheet form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

Privacy Act Statement

The Privacy Act of 1974 (P.L. 93-579) requires that you be given certain information in connection with your submission of the attached form related to a patent application or patent. Accordingly, pursuant to the requirements of the Act, please be advised that: (1) the general authority for the collection of this information is 35 U.S.C. 2(b)(2); (2) furnishing of the information solicited is voluntary; and (3) the principal purpose for which the information is used by the U.S. Patent and Trademark Office is to process and/or examine your submission related to a patent application or patent. If you do not furnish the requested information, the U.S. Patent and Trademark Office may not be able to process and/or examine your submission, which may result in termination of proceedings or abandonment of the application or expiration of the patent.

The information provided by you in this form will be subject to the following routine uses:

1. The information on this form will be treated confidentially to the extent allowed under the Freedom of Information Act (5 U.S.C. 552) and the Privacy Act (5 U.S.C. 552a). Records from this system of records may be disclosed to the Department of Justice to determine whether the Freedom of Information Act requires disclosure of these records.

2. A record from this system of records may be disclosed, as a routine use, in the course of presenting evidence to a court, magistrate, or administrative tribunal, including disclosures to opposing counsel in the course of settlement negotiations.

3. A record in this system of records may be disclosed, as a routine use, to a Member of Congress submitting a request involving an individual, to whom the record pertains, when the individual has requested assistance from the Member with respect to the subject matter of the record.

4. A record in this system of records may be disclosed, as a routine use, to a contractor of the Agency having need for the information in order to perform a contract. Recipients of information shall be required to comply with the requirements of the Privacy Act of 1974, as amended, pursuant to 5 U.S.C. 552a(m).

5. A record related to an International Application filed under the Patent Cooperation Treaty in this system of records may be disclosed, as a routine use, to the International Bureau of the World Intellectual Property Organization, pursuant to the Patent Cooperation Treaty.

6. A record in this system of records may be disclosed, as a routine use, to another federal agency for purposes of National Security review (35 U.S.C. 181) and for review pursuant to the Atomic Energy Act (42 U.S.C. 218(c)).

7. A record from this system of records may be disclosed, as a routine use, to the Administrator, General Services, or his/her designee, during an inspection of records conducted by GSA as part of that agency's responsibility to recommend improvements in records management practices and programs, under authority of 44 U.S.C. 2904 and 2906. Such disclosure shall be made in accordance with the GSA regulations governing inspection of records for this purpose, and any other relevant (i.e., GSA or Commerce) directive. Such disclosure shall not be used to make determinations about individuals.

8. A record from this system of records may be disclosed, as a routine use, to the public after either publication of the application pursuant to 35 U.S.C. 122(b) or issuance of a patent pursuant to 35 U.S.C. 151. Further, a record may be disclosed, subject to the limitations of 37 CFR 1.14, as a routine use, to the public if the record was filed in an application which became abandoned or in which the proceedings were terminated and which application is referenced by either a published application, an application open to public inspections or an issued patent.

9. A record from this system of records may be disclosed, as a routine use, to a Federal, State, or local law enforcement agency, if the USPTO becomes aware of a violation or potential violation of law or regulation.

Electronic Patent Application Fee Transmittal

Application Number:	
Filing Date:	
Title of Invention:	Mobile Device
First Named Inventor/Applicant Name:	Denise Burton
Filer:	Tadd Fjelp Wilson/Amy Duarte
Attorney Docket Number:	6583

Filed as Large Entity

Provisional Filing Fees

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Basic Filing:				
Provisional application filing	1005	1	250	250
Pages:				
Prov. Appl Size fee per 50 sheets >100	1085	6	310	1860
Claims:				
Miscellaneous-Filing:				
Petition:				
Patent-Appeals-and-Interference:				
Post-Allowance-and-Post-Issuance:				

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Extension-of-Time:				
Miscellaneous:				
			Total in USD ($)	**2110**

Electronic Acknowledgement Receipt

EFS ID:	11063033
Application Number:	61539884
International Application Number:	
Confirmation Number:	2198
Title of Invention:	Mobile Device
First Named Inventor/Applicant Name:	Denise Burton
Customer Number:	22442
Filer:	Tadd Fjelp Wilson/Amy Duarte
Filer Authorized By:	Tadd Fjelp Wilson
Attorney Docket Number:	6583
Receipt Date:	27-SEP-2011
Filing Date:	
Time Stamp:	19:22:21
Application Type:	Provisional

Payment information:

Submitted with Payment	yes
Payment Type	Deposit Account
Payment was successfully received in RAM	$2110
RAM confirmation Number	6520
Deposit Account	191970
Authorized User	

The Director of the USPTO is hereby authorized to charge indicated fees and credit any overpayment as follows:

Charge any Additional Fees required under 37 C.F.R. Section 1.16 (National application filing, search, and examination fees)

Charge any Additional Fees required under 37 C.F.R. Section 1.17 (Patent application and reexamination processing fees)

Charge any Additional Fees required under 37 C.F.R. Section 1.19 (Document supply fees)

Charge any Additional Fees required under 37 C.F.R. Section 1.21 (Miscellaneous fees and charges)

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Specification	6583_Provisional_Application_Final.pdf	6080715 2c113e4d46b3618e9cc150e33596ea1888cb948b	no	487
Warnings:					
Information:					
2	Application Data Sheet	ADS.pdf	1431869 2a3c6df6a1e962cb6d8ed97b1b6f26e381a5c732	no	11
Warnings:					
Information:					
3	Fee Worksheet (SB06)	fee-info.pdf	31069 079f8c0dfb6bf6ae28ccc6e8b28705a92c7ee6e2	no	2
Warnings:					
Information:					
		Total Files Size (in bytes):	7543653		

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111

If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371

If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office

If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NUMBER	FILING or 371(c) DATE	GRP ART UNIT	FIL FEE REC'D	ATTY.DOCKET.NO	TOT CLAIMS	IND CLAIMS
61/539,884	09/27/2011		2110	6583		

CONFIRMATION NO. 2198

22442
SHERIDAN ROSS PC
1560 BROADWAY
SUITE 1200
DENVER, CO 80202

FILING RECEIPT

OC000000050333063

Date Mailed: 10/13/2011

Receipt is acknowledged of this provisional patent application. It will not be examined for patentability and will become abandoned not later than twelve months after its filing date. Any correspondence concerning the application must include the following identification information: the U.S. APPLICATION NUMBER, FILING DATE, NAME OF APPLICANT, and TITLE OF INVENTION. Fees transmitted by check or draft are subject to collection. Please verify the accuracy of the data presented on this receipt. **If an error is noted on this Filing Receipt, please submit a written request for a Filing Receipt Correction. Please provide a copy of this Filing Receipt with the changes noted thereon. If you received a "Notice to File Missing Parts" for this application, please submit any corrections to this Filing Receipt with your reply to the Notice. When the USPTO processes the reply to the Notice, the USPTO will generate another Filing Receipt incorporating the requested corrections**

Applicant(s)

Denise Burton, Austin, TX;
Ron Cassar, Burlington, CANADA;
Tong Chen, Markham, CANADA;
Misty Cripps, Austin, TX;
Robert Csiki, Markham, CANADA;
Alexander de Paz, Burlington, CANADA;
Brett Bernard Faulk, Cypress, TX;
Jared L. Ficklin, Austin, TX;
Eric Freedman, Austin, TX;
Martin Gimpl, Helsinki, FINLAND;
Volodimir Felixovich Lemberg, Etobicoke, CANADA;
Maxim Marintchenko, Vaughan, CANADA;
Eduardo Diego Torres Milano, Burlington, CANADA;
Paul Edward Reeves, Oakville, CANADA;
Rodney Wayne Schrock, San Diego, CA;
Mohammed Selim, Oakville, CANADA;
Sanjiv Sirpal, Oakville, CANADA;
Nikhil Swaminathan, Toronto, CANADA;
John Steven Visosky, Gormley, CANADA;
Suxiang Wei, Mississauga, CANADA;
Gregg Wygonik, Austin, TX;
Miroslaw Zaremski, Richmond Hill, CANADA;

Power of Attorney:
Tadd Wilson--54544

If Required, Foreign Filing License Granted: 10/12/2011

The country code and number of your priority application, to be used for filing abroad under the Paris Convention, is **US 61/539,884**

Projected Publication Date: None, application is not eligible for pre-grant publication

Non-Publication Request: No

Early Publication Request: No
Title

MOBILE DEVICE

PROTECTING YOUR INVENTION OUTSIDE THE UNITED STATES

Since the rights granted by a U.S. patent extend only throughout the territory of the United States and have no effect in a foreign country, an inventor who wishes patent protection in another country must apply for a patent in a specific country or in regional patent offices. Applicants may wish to consider the filing of an international application under the Patent Cooperation Treaty (PCT). An international (PCT) application generally has the same effect as a regular national patent application in each PCT-member country. The PCT process **simplifies** the filing of patent applications on the same invention in member countries, but **does not result** in a grant of "an international patent" and does not eliminate the need of applicants to file additional documents and fees in countries where patent protection is desired.

Almost every country has its own patent law, and a person desiring a patent in a particular country must make an application for patent in that country in accordance with its particular laws. Since the laws of many countries differ in various respects from the patent law of the United States, applicants are advised to seek guidance from specific foreign countries to ensure that patent rights are not lost prematurely.

Applicants also are advised that in the case of inventions made in the United States, the Director of the USPTO must issue a license before applicants can apply for a patent in a foreign country. The filing of a U.S. patent application serves as a request for a foreign filing license. The application's filing receipt contains further information and guidance as to the status of applicant's license for foreign filing.

Applicants may wish to consult the USPTO booklet, "General Information Concerning Patents" (specifically, the section entitled "Treaties and Foreign Patents") for more information on timeframes and deadlines for filing foreign patent applications. The guide is available either by contacting the USPTO Contact Center at 800-786-9199, or it can be viewed on the USPTO website at http://www.uspto.gov/web/offices/pac/doc/general/index.html.

For information on preventing theft of your intellectual property (patents, trademarks and copyrights), you may wish to consult the U.S. Government website, http://www.stopfakes.gov. Part of a Department of Commerce initiative, this website includes self-help "toolkits" giving innovators guidance on how to protect intellectual property in specific countries such as China, Korea and Mexico. For questions regarding patent enforcement issues, applicants may call the U.S. Government hotline at 1-866-999-HALT (1-866-999-4158).

LICENSE FOR FOREIGN FILING UNDER

Title 35, United States Code, Section 184

Title 37, Code of Federal Regulations, 5.11 & 5.15

GRANTED

The applicant has been granted a license under 35 U.S.C. 184, if the phrase "IF REQUIRED, FOREIGN FILING LICENSE GRANTED" followed by a date appears on this form. Such licenses are issued in all applications where the conditions for issuance of a license have been met, regardless of whether or not a license may be required as set forth in 37 CFR 5.15. The scope and limitations of this license are set forth in 37 CFR 5.15(a) unless an earlier license has been issued under 37 CFR 5.15(b). The license is subject to revocation upon written notification. The date indicated is the effective date of the license, unless an earlier license of similar scope has been granted under 37 CFR 5.13 or 5.14.

This license is to be retained by the licensee and may be used at any time on or after the effective date thereof unless it is revoked. This license is automatically transferred to any related applications(s) filed under 37 CFR 1.53(d). This license is not retroactive.

The grant of a license does not in any way lessen the responsibility of a licensee for the security of the subject matter as imposed by any Government contract or the provisions of existing laws relating to espionage and the national security or the export of technical data. Licensees should apprise themselves of current regulations especially with respect to certain countries, of other agencies, particularly the Office of Defense Trade Controls, Department of State (with respect to Arms, Munitions and Implements of War (22 CFR 121-128)); the Bureau of Industry and Security, Department of Commerce (15 CFR parts 730-774); the Office of Foreign AssetsControl, Department of Treasury (31 CFR Parts 500+) and the Department of Energy.

NOT GRANTED

No license under 35 U.S.C. 184 has been granted at this time, if the phrase "IF REQUIRED, FOREIGN FILING LICENSE GRANTED" DOES NOT appear on this form. Applicant may still petition for a license under 37 CFR 5.12, if a license is desired before the expiration of 6 months from the filing date of the application. If 6 months has lapsed from the filing date of this application and the licensee has not received any indication of a secrecy order under 35 U.S.C. 181, the licensee may foreign file the application pursuant to 37 CFR 5.15(b).