US 20020191029A1

(19) United States

(12) Patent Application Publication

(10) Pub. No.: US 2002/0191029 A1

(43) Pub. Date: Dec. 19, 2002

Gillespie et al.

(54) TOUCH SCREEN WITH USER INTERFACE ENHANCEMENT

(75) Inventors: **David W. Gillespie**, Los Gatos, CA (US); **Ray Trent**, San Jose, CA (US); **Andrew C. Hsu**, Belmont, CA (US); **Leslie R. Grate**, Waltham, MA (US)

Correspondence Address:
SIERRA PATENT GROUP, LTD.
P O BOX 6149
STATELINE, NV 89449 (US)

(73) Assignee: **Synaptics, Inc.**

(21) Appl. No.: **10/125,066**

(22) Filed: **Apr. 17, 2002**

Related U.S. Application Data

(60) Provisional application No. 60/291,694, filed on May 16, 2001.

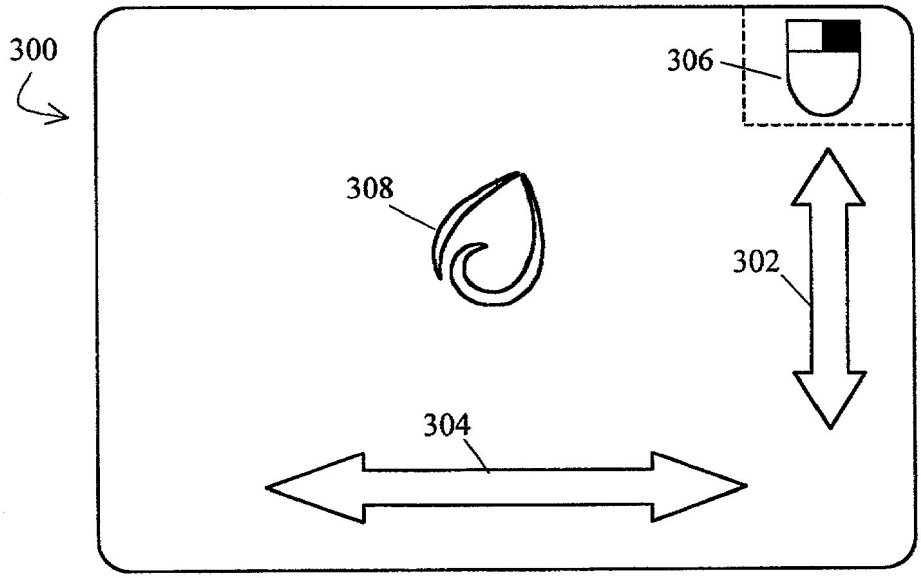
Publication Classification

(51) **Int. Cl.**[7] **G09G 5/00**

(52) **U.S. Cl.** **345/810**

(57) **ABSTRACT**

The present invention is a graphical user interface in a computing device having a processor running an operating system and a display. The graphical user interface comprises a touch screen and a driver coupling the touch screen to the operating system. The driver can display a plurality of icons on the touch screen, or a plurality of screen images having at least one icon, with each of the icons associated with operations on the display and/or the touch screen. Other embodiments include the touch screen having unactivated and activated states, as well as the presence of an application programming interface that enables an application to display at least one image on the touch screen.



Bose Exhibit 1015
Bose v. IngenioSpec

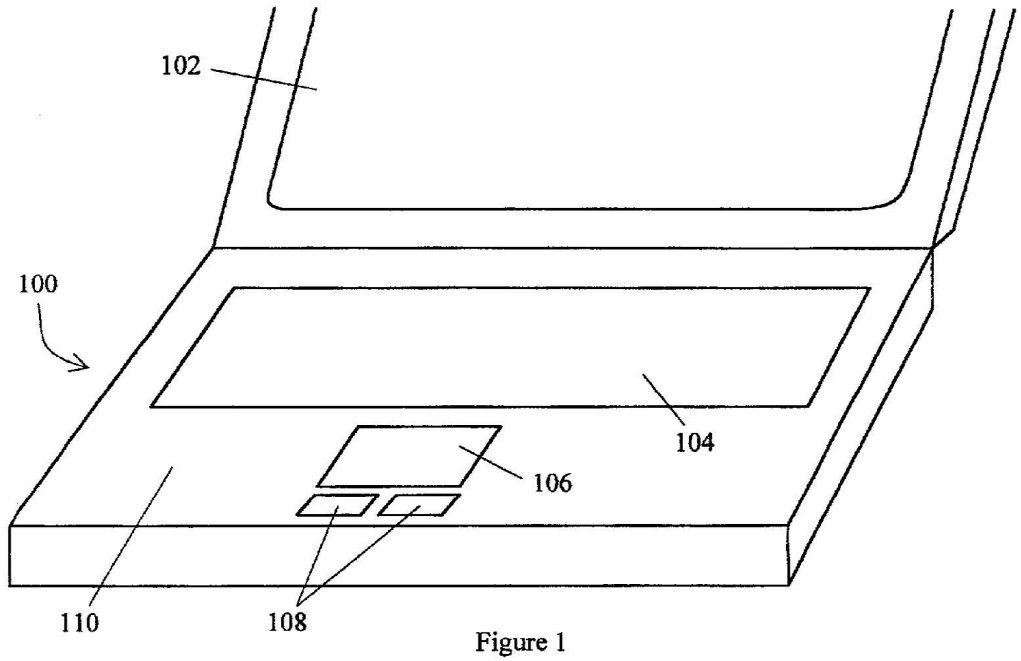


Figure 1

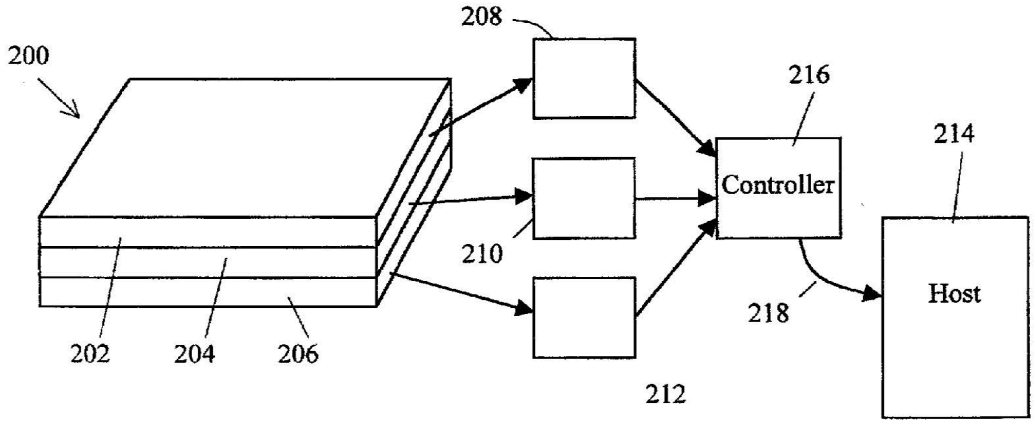


Figure 2

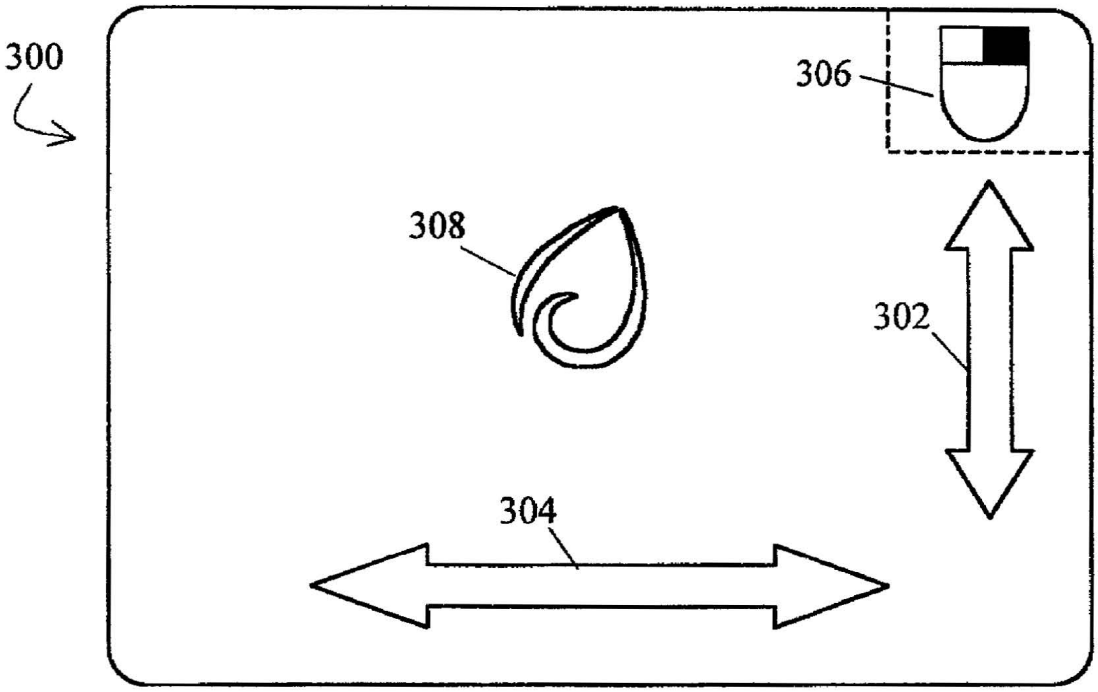


Figure 3

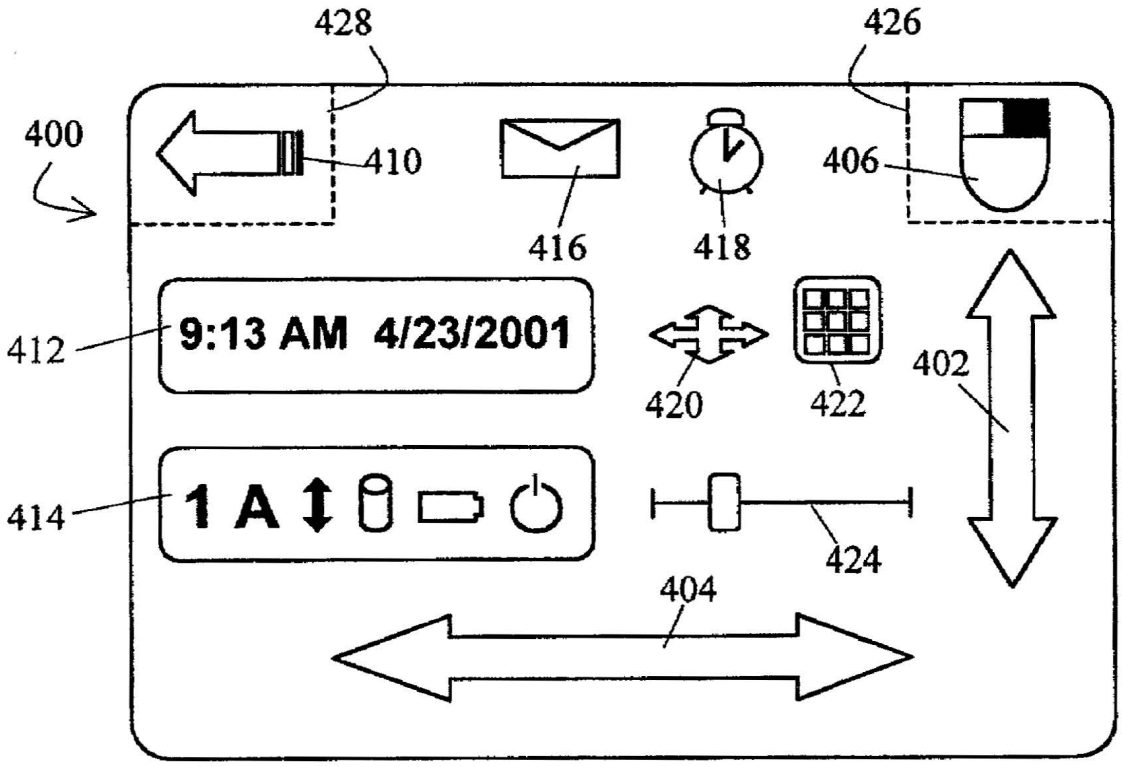


Figure 4

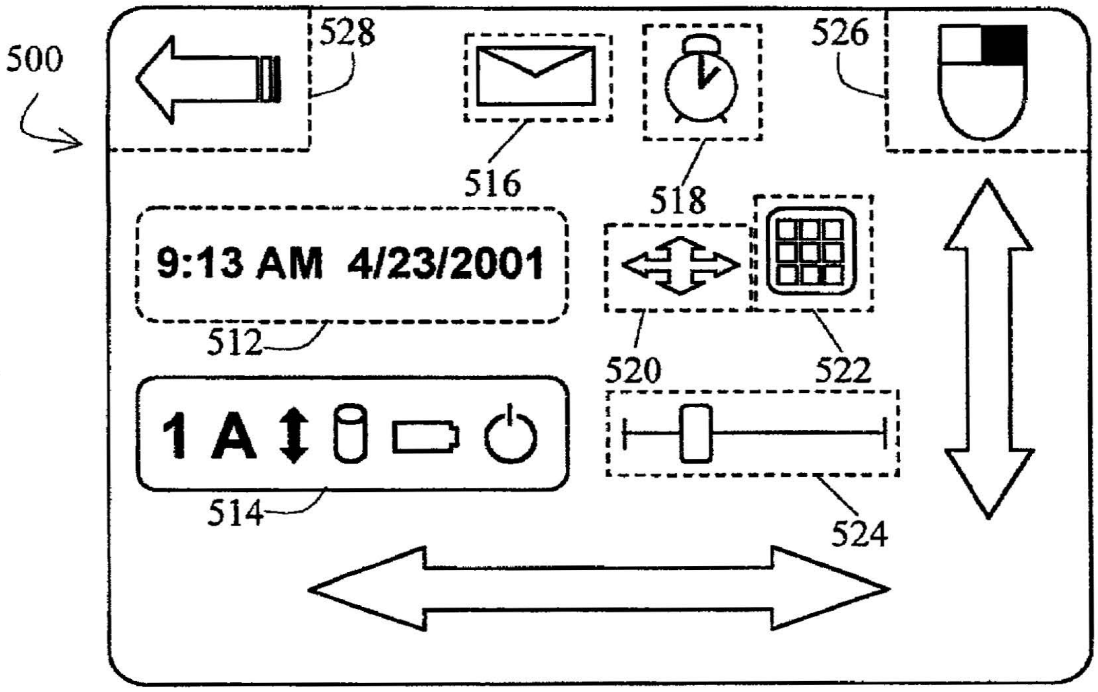


Figure 5

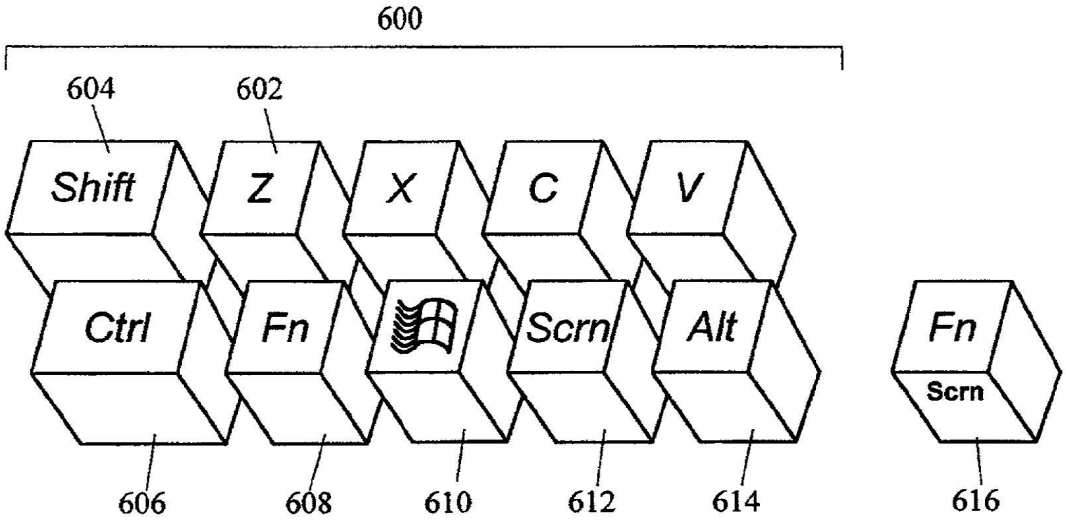


Figure 6A

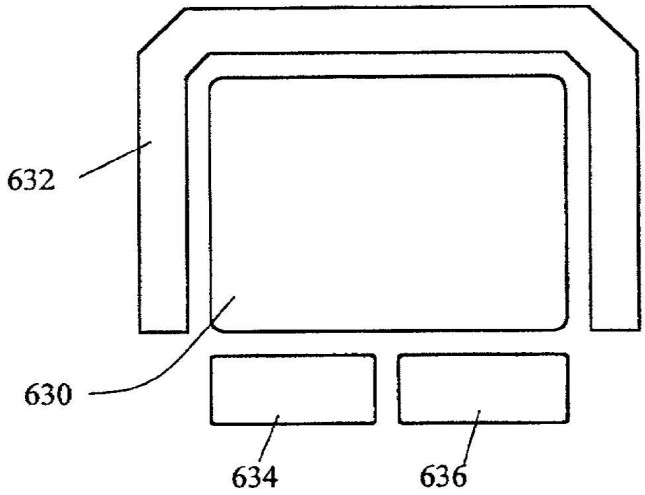


Figure 6B

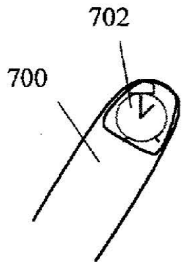


Figure 7A

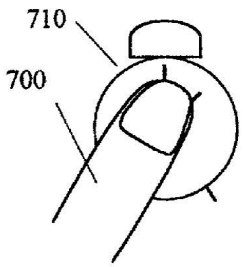


Figure 7B

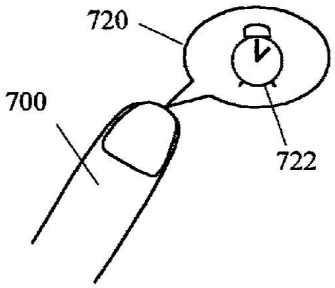


Figure 7C

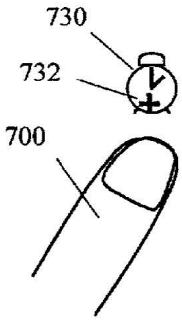


Figure 7D

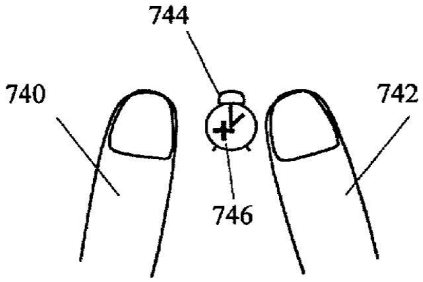


Figure 7E

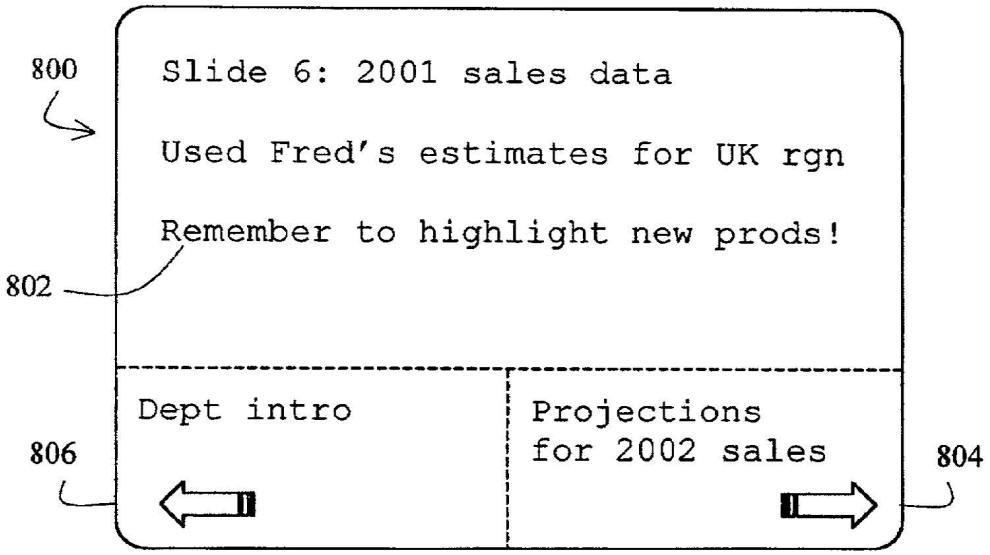


Figure 8A

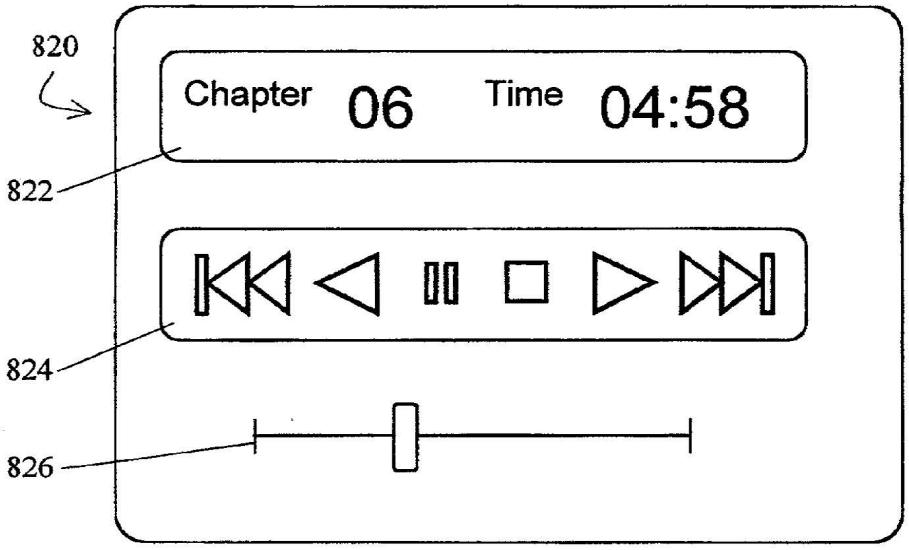


Figure 8B

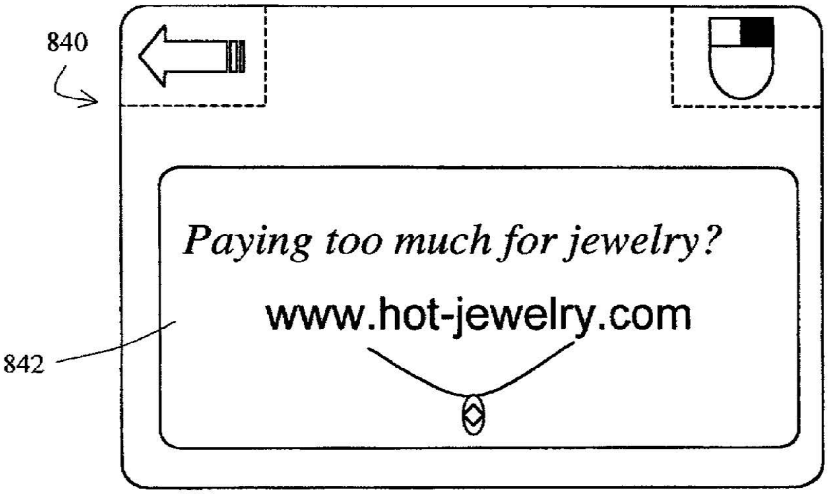


Figure 8C

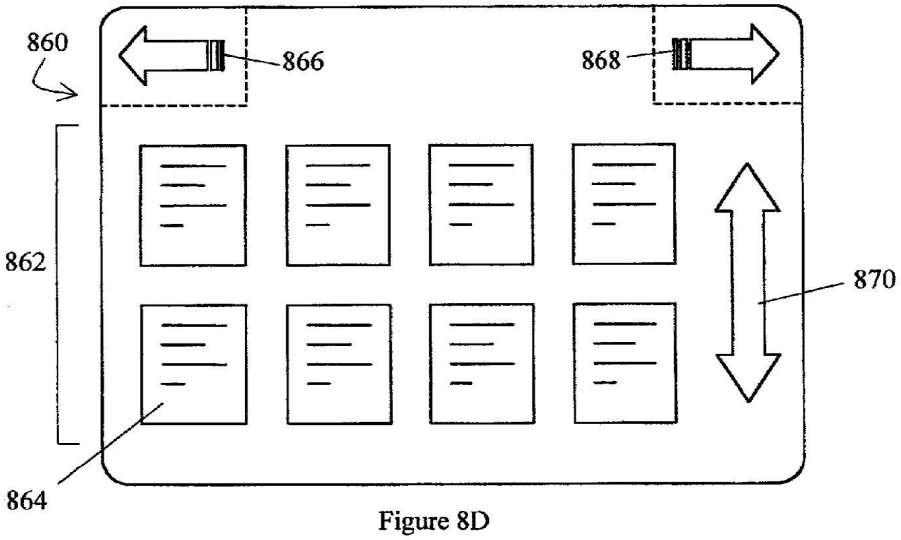


Figure 8D

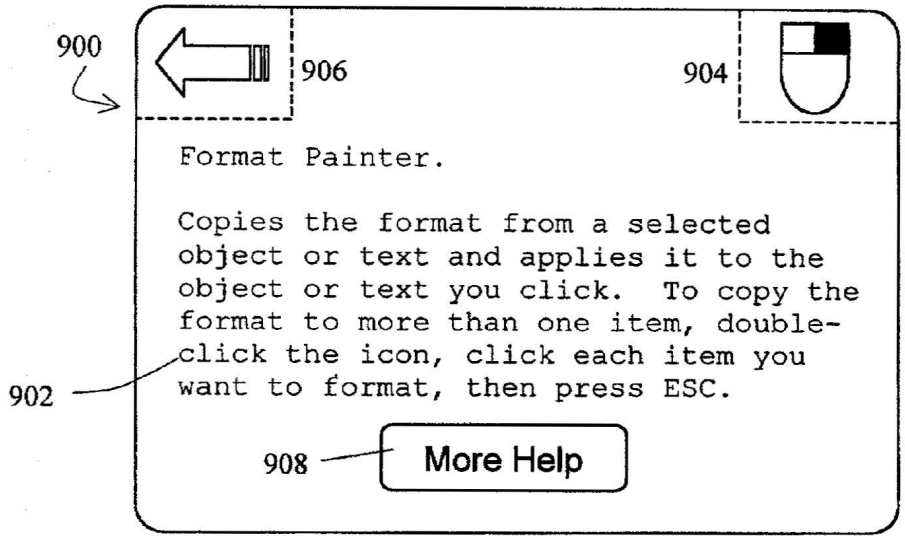


Figure 9

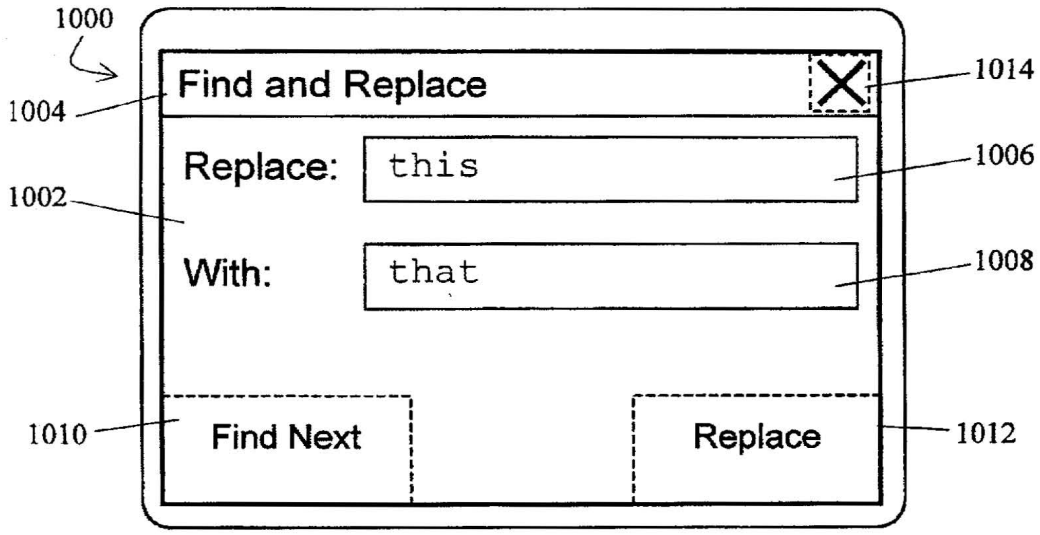


Figure 10A

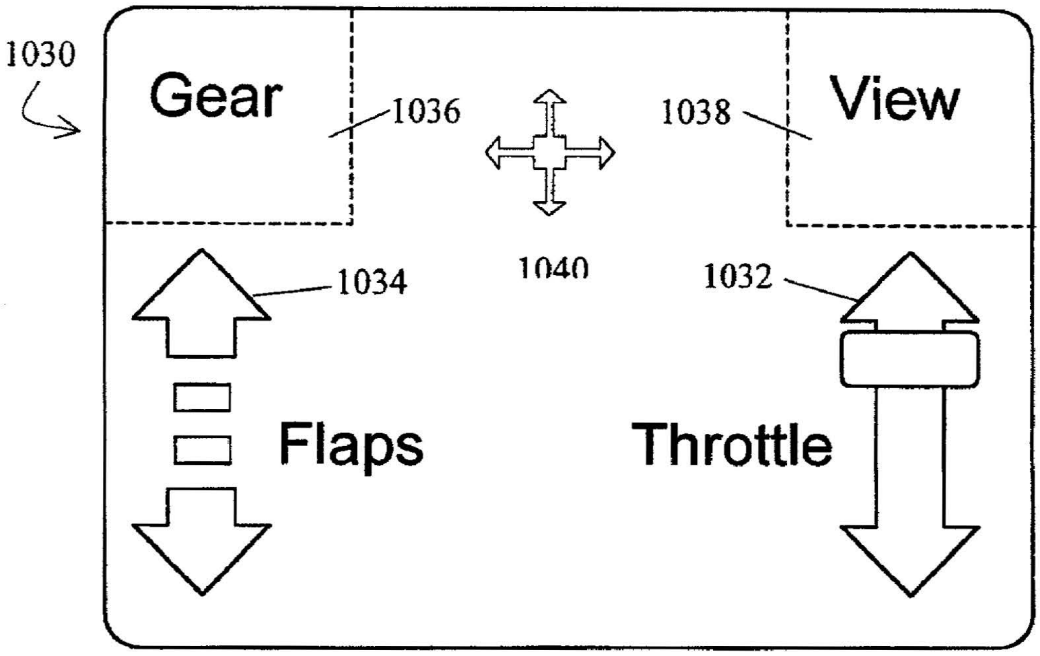


Figure 10B

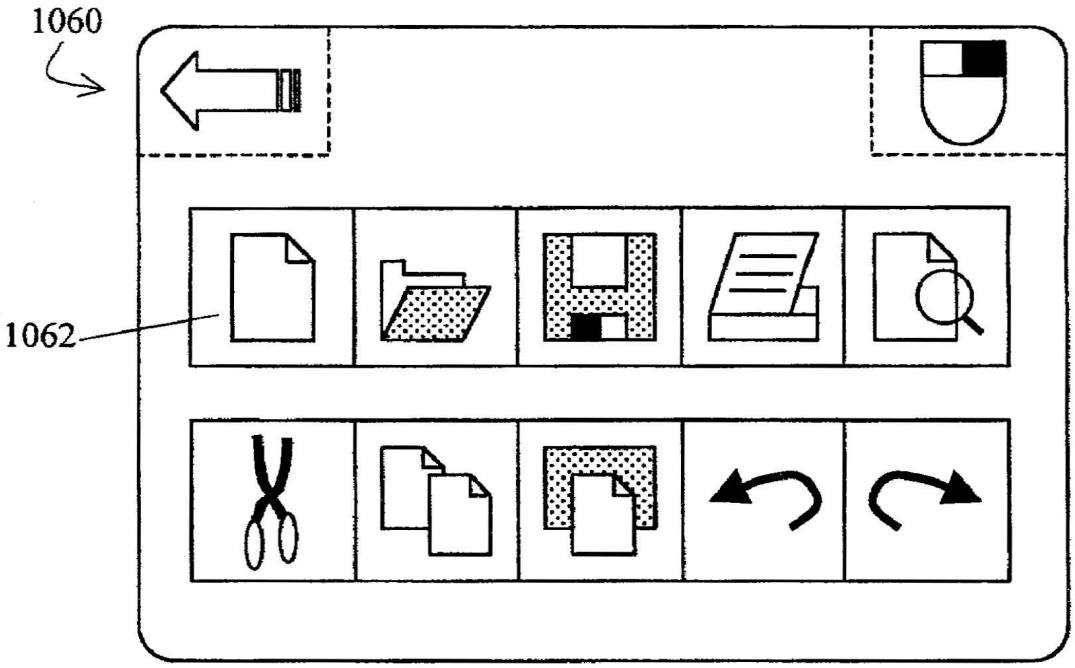


Figure 10C

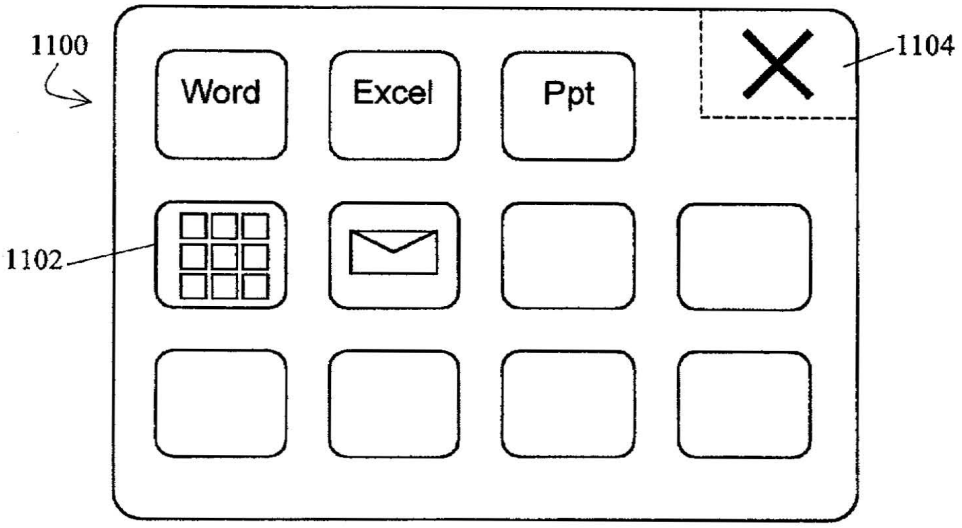


Figure 11

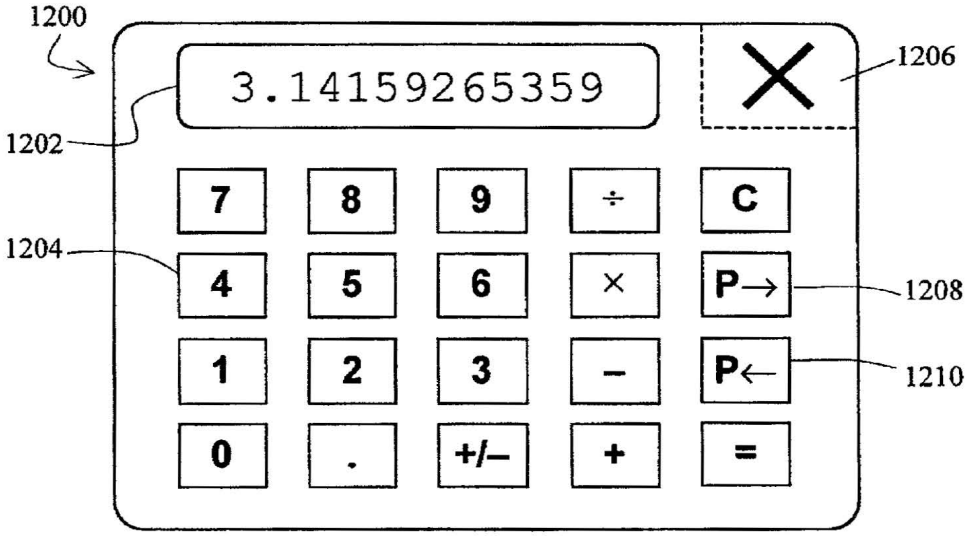


Figure 12

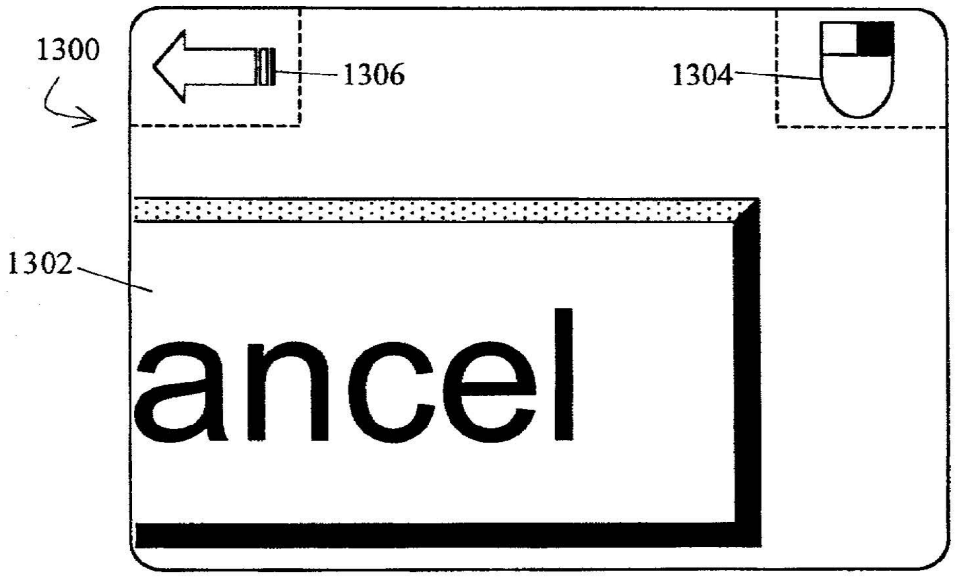


Figure 13A

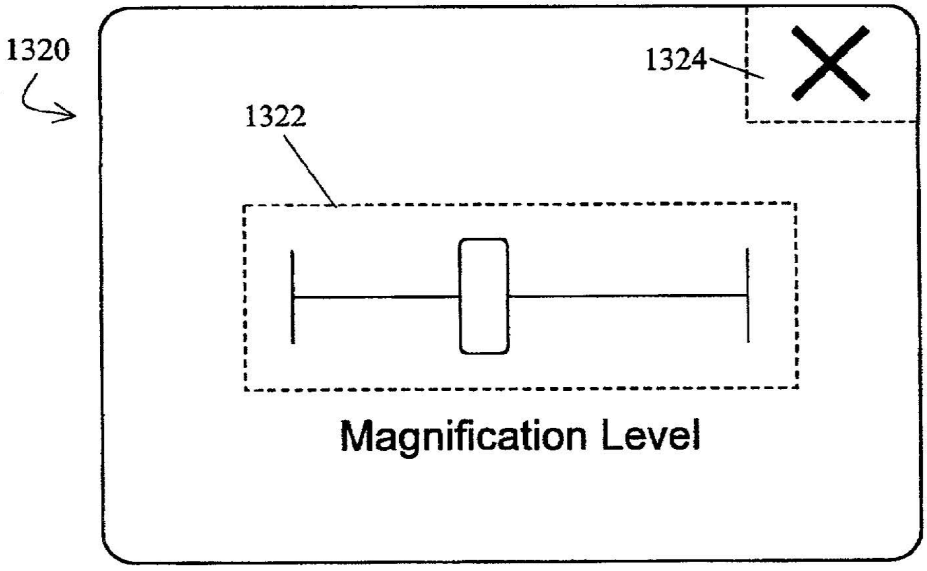


Figure 13B

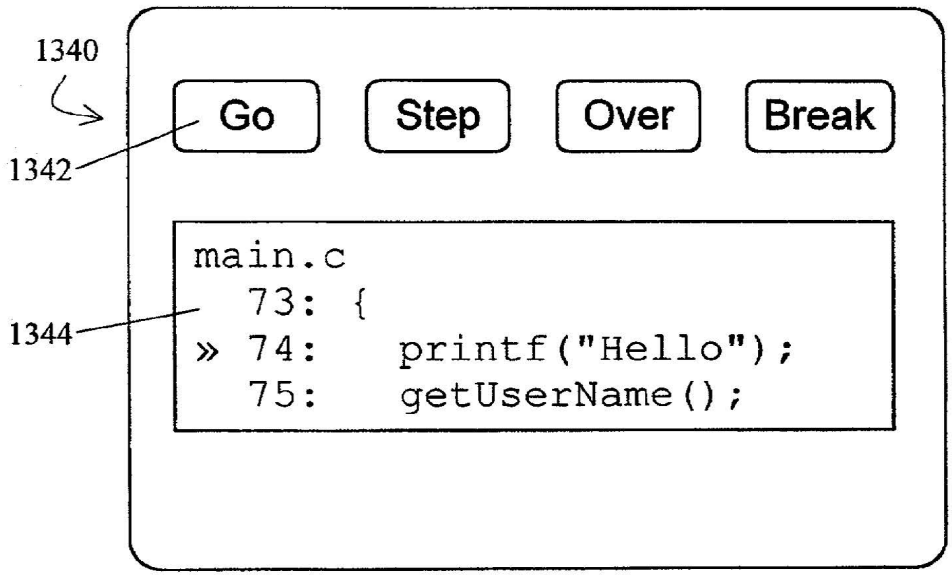


Figure 13C

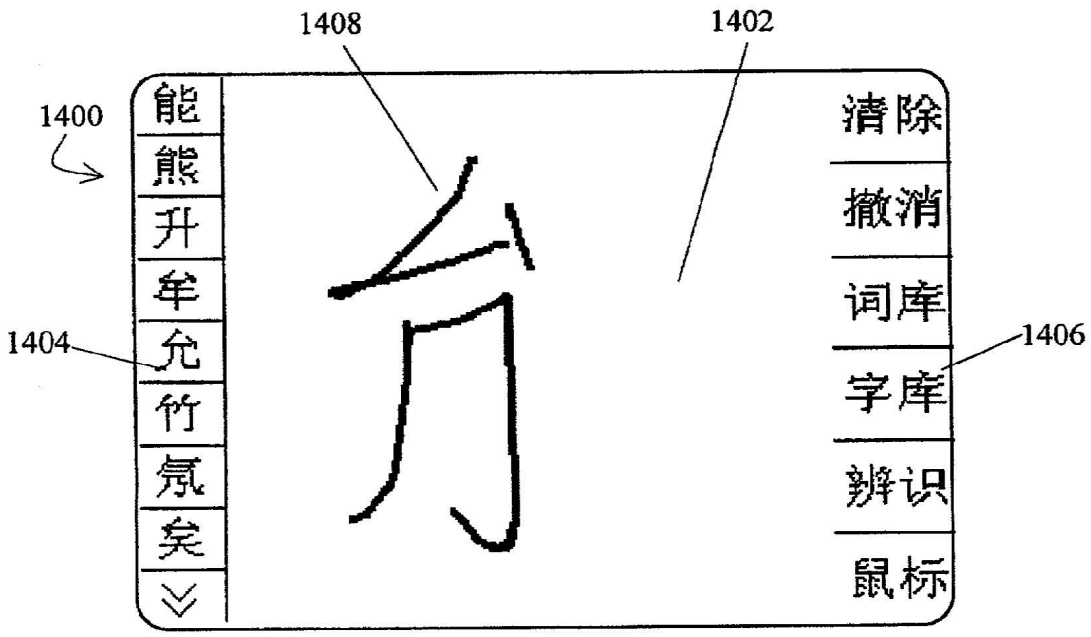


Figure 14

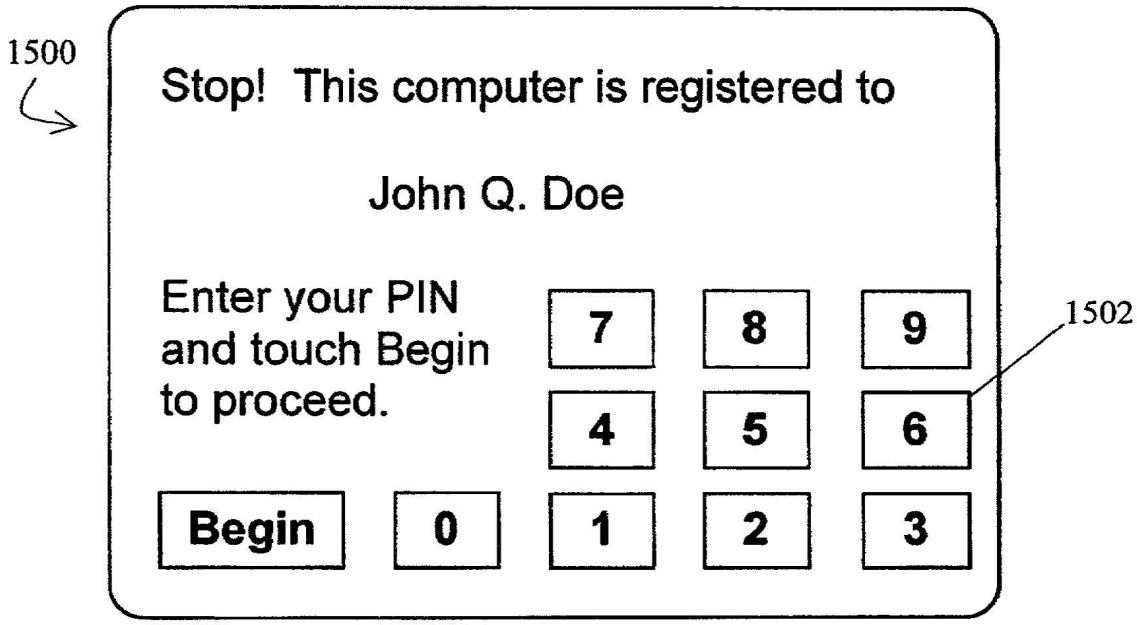


Figure 15A

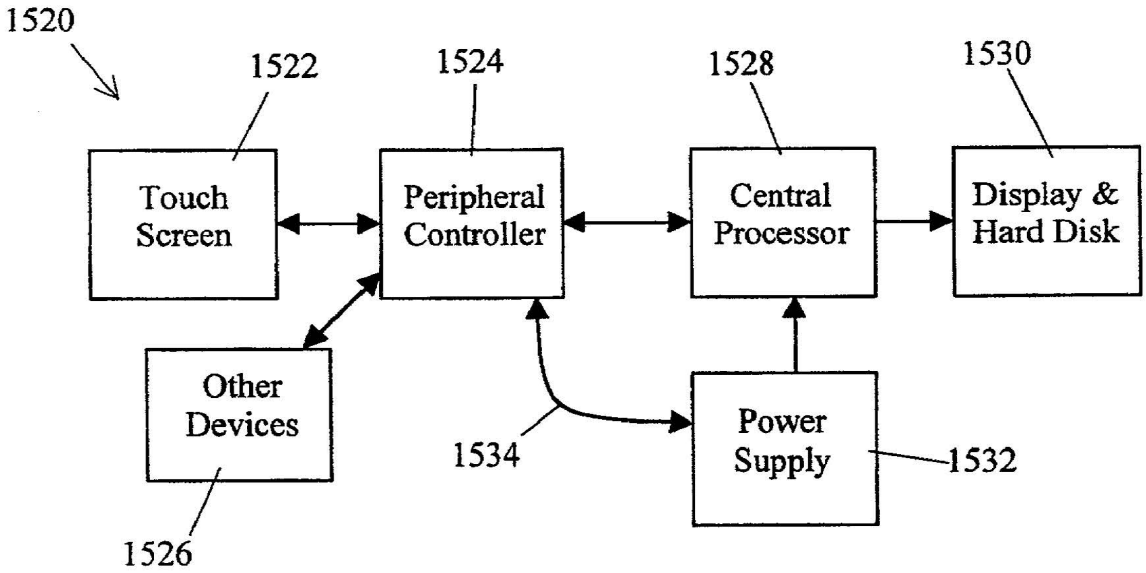


Figure 15B

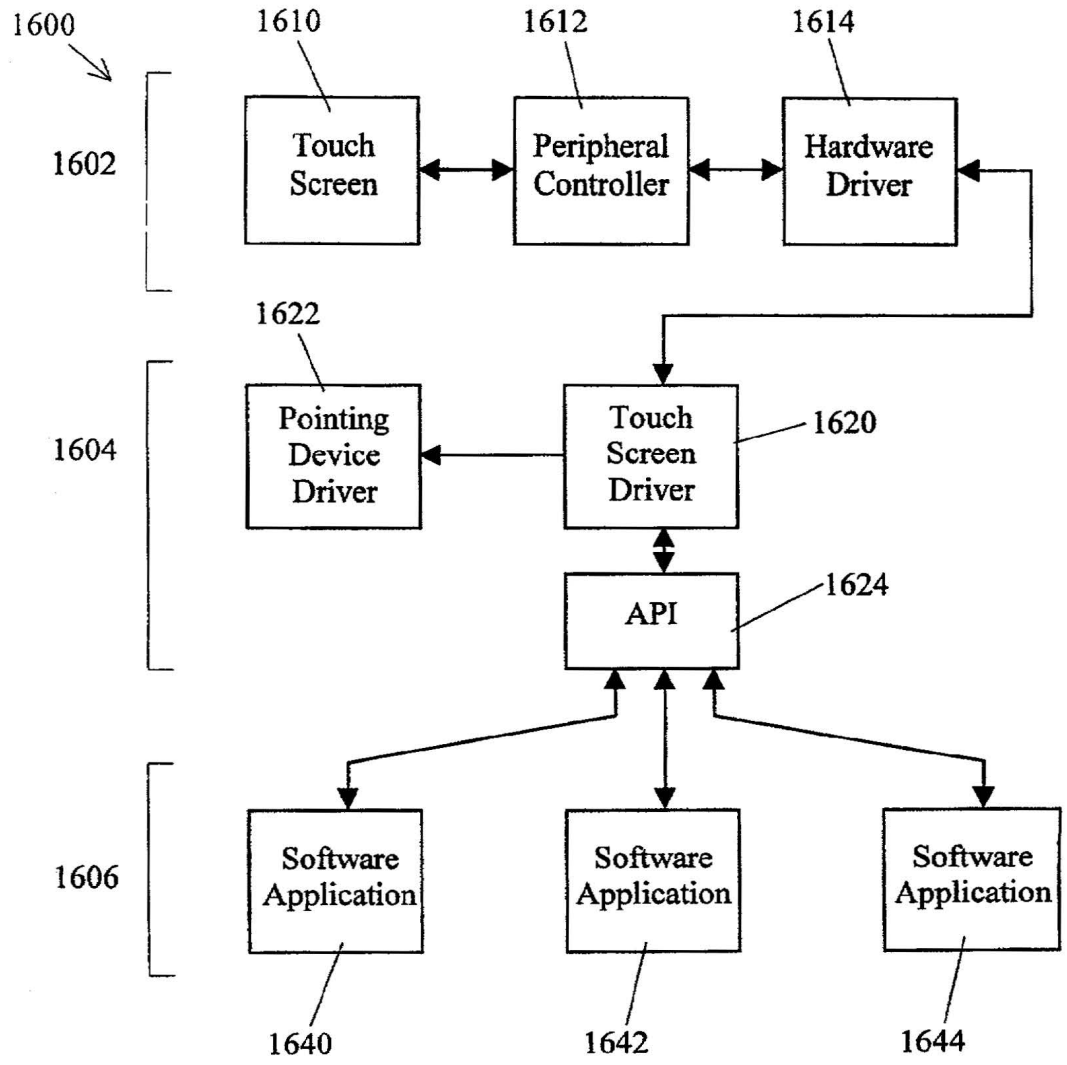

1600
1610
1612
1614
1602
Touch Screen
Peripheral Controller
Hardware Driver
1622
1604
Pointing Device Driver
Touch Screen Driver
1620
API
1624
1606
Software Application
Software Application
Software Application
1640
1642
1644

Figure 16

TOUCH SCREEN WITH USER INTERFACE ENHANCEMENT

PRIORITY TO RELATED APPLICATIONS

[0001] The present application claims priority to U.S. Patent Provisional Application Serial No. 60/291,694, entitled "Touch Screen with User Interface Enhancement", filed on May 16, 2001, which is incorporated herein in its entirety.

BACKGROUND

[0002] The present invention relates to computer interface devices, and more particularly, to a computer touch pad with integrated display device, and enhancements to the portable computer user interface employing same.

[0003] Touch pads are widely used in computer applications, particularly as pointing devices in portable computers. In typical usage, the touch pad is a featureless, finger sensitive surface in a rectangular opening of the palm rest of the computer. The touch pad serves solely as an input device for the computer. The touch pad functions primarily as a cursor pointing device, but some touch pads offer additional functions.

[0004] For example, U.S. Pat. No. 5,543,591 to Gillespie et al. discloses a typical prior art touch pad sensor in which finger tapping gestures in designated regions of the touch surface invoke special commands on the computer. U.S. Pat. No. 5,943,052 to Allen et al. discloses a touch pad in which finger motions in designated regions invoke a scrolling command. These tap regions and scrolling regions have proven useful to expert users but confusing to novice users as the regions are invisible to the eye but different in behavior. Marking the regions with screen-printed icons on the opaque sensor surface can help, but it can also lead to greater confusion if the regions are software configurable.

[0005] A further disadvantage of prior art touch pads is that they use up a significant fraction of the surface area of the computer for a single dedicated input function. Other pointing devices such as isometric joysticks (see, e.g., U.S. Pat. No. 5,521,596 to Selker et al) and force sensing keys (see, e.g., U.S. Pat. No. 4,680,577 to Straayer et al) have been proposed as compact alternatives, but these devices are not as expressive or as easy to use as touch pads.

[0006] Touch screens are also well known in the art. One example of a touch screen is disclosed in U.S. Pat. No. 4,806,709 to Blair. In typical use, the main display screen of a computer is overlaid with or implemented as a touch sensitive input device. This eliminates the need to dedicate separate parts of the surface of the computer for input and output. If the touch screen serves as the main pointing device of the computer, pointing is accomplished by a direct mapping from finger position to selection of a point on the screen beneath the finger. This direct mapping makes touch screens easy to understand and use. However, touch screens are impractical for everyday use as the main display of a computer because the user's arm tires from being continuously held up to touch the screen. If the touch screen is laid flat to avoid arm wear, the arm tends to rest on the touch-sensing surface and, with many touch sensing technologies, this disrupts the ability to sense the finger. Touch screens the size of a main computer display may also be prohibitively bulky or expensive for use in applications that do not require them.

[0007] A transparent touch pad suitable for placement over a display such as an LCD screen has been developed and is disclosed and claimed in co-pending U.S. patent application Ser. No. 09/415,481, filed Oct. 8, 1999, assigned to the same assignee as the present invention. This application discloses a touch screen having the small size and low cost of a conventional touch pad for portable computers and notes that the touch pad and display could be included in a personal computer to enhance the user interface in various ways, but it does not disclose details of the software implementation, nor how such a device can simultaneously function as the pointing device of the computer, nor how this arrangement enhances the user interface.

SUMMARY

[0008] The drawbacks and disadvantages of the prior art are overcome by the touch screen with user interface enhancement.

[0009] The present invention discloses a graphical user interface in a computing device having a processor running an operating system and a display. The graphical user interface comprises a touch screen and a driver coupling the touch screen to the operating system. The driver displays a plurality of icons on the touch screen with at least one of the icons identifying at least one region on the touch screen that will cause an action on the display and not on the touch screen in response to contact by an object on the region.

[0010] The present invention also discloses another embodiment of a graphical user interface in a computing device having a processor running an operating system and a display. The graphical user interface comprises a touch screen, having an unactivated state and an activated state, and a driver coupling the touch screen to the operating system. The driver displays a plurality of icons on the touch screen with at least one of the icons identifying at least one region on the touch screen that will cause an action on the display and not on the touch screen in response to contact by an object on the region.

[0011] Lastly, the present invention discloses another embodiment of a graphical user interface in a computing device having a processor running an operating system and a display. The graphical user interface comprises a touch screen and a driver coupling the touch screen to the operating system. The driver displays a plurality of icons on the touch screen with at least one of the icons identifying at least one region on the touch screen that will cause an action on the display and not on the touch screen in response to contact by an object on the region. The driver has an application programming interface that enables an application to display at least one image on the touch screen.

BRIEF DESCRIPTION OF THE DRAWING FIGURES

[0012] Referring now to the figures, wherein like elements are numbered alike:

[0013] **FIG. 1** is a diagram showing a notebook computer system with main display, keyboard, and touch screen;

[0014] **FIG. 2** is a diagram showing an illustrative embodiment of a touch screen in greater detail;

[0015] **FIG. 3** is a diagram illustrating an example default image for use when the touch screen is operating as a conventional touch pad;

[0016] **FIG. 4** is a diagram illustrating an example of a first "iconic" usage mode of the touch screen;

[0017] **FIG. 5** is a diagram illustrating the touch screen image of **FIG. 4** modified to indicate the activated state of the touch screen using a dashed line around each icon that is touch-sensitive in the activated state;

[0018] **FIG. 6A** is a diagram illustrating a portion of the keyboard featuring several keys;

[0019] **FIG. 6B** is a diagram illustrating one possible arrangement of a special touch sensitive region or second touch sensor could be provided that activates the touch screen when touched;

[0020] **FIG. 7A** is a diagram illustrating small icons that may be smaller than a finger and may be completely obscured by the finger when the finger touches them;

[0021] **FIG. 7B through 7E** illustrate several mechanisms to eliminate the problem of obscuring small icons;

[0022] **FIGS. 8A through 8D** are diagrams illustrating use of a small control panel on the touch screen associated with an application, reserving the entire main display for visual data associated with the application;

[0023] **FIG. 9** is a diagram showing an example use of the touch screen to display subsidiary help text;

[0024] **FIG. 10A** is a diagram illustrating employment of the touch screen to display a find/replace dialog on the touch screen, leaving the main display free to display a document unobstructed;

[0025] **FIG. 10B** is a diagram illustrating use of the touch screen to act as a joystick emulator while displaying the control layout established by the game, leaving the main display free to display game graphics unobstructed;

[0026] **FIG. 10C** is a diagram illustrating an example in which a touch screen image includes icons drawn from a typical toolbar, leaving the main display free to display document or an image unobstructed;

[0027] **FIG. 11** is a diagram illustrating a pop-up image including various icons representing commonly used tools and software applications on the computer;

[0028] **FIG. 12** is a diagram illustrating a pop-up calculator application that operates entirely within the touch screen;

[0029] **FIG. 13A and 13B** are diagrams illustrating different features of a magnifier as a pop-up image on a touch screen, leaving the main display undisturbed;

[0030] **FIG. 13C** is a diagram illustrating a debugger implemented as a pop-up application on a touch screen, providing a secondary debugging display with no extra cost or bulk;

[0031] **FIG. 14** is a diagram illustrating an example of an ideographic handwriting entry system on a touch screen in which a handwriting entry area responds to finger touch to enter an ideographic character;

[0032] **FIG. 15A** is a diagram illustrating use of a touch screen as a user interface device for a computer security interlock;

[0033] **FIG. 15B** is a diagram illustrating an exemplary hardware architecture for implementing the computer security interlock of **FIG. 15A**; and

[0034] **FIG. 16** is a diagram illustrating an exemplary software architecture for a touch screen.

DETAILED DESCRIPTION

[0035] Those of ordinary skill in the art will realize that the following description of the present invention is illustrative only and not in any way limiting. Other embodiments of the invention will readily suggest themselves to such skilled persons.

[0036] **FIG. 1** illustrates a notebook computer system **100** with main display **102** and keyboard **104**. Touch screen **106** is mounted in palm rest **110**. The touch screen is typically equipped with left and right "mouse" buttons **108**. Touch screen **106** is integrated into computer system **100** in a similar way as a touch pad would be in a prior art computer. Touch screen **106** will usually be located in the palm rest as shown in **FIG. 1**, but other locations are equally applicable, such as above the keyboard, adjacent to the keyboard or main display, or located in a separate enclosure connected by cable or wireless link to the computer. Although touch screen **106** usually replaces the conventional touch pad of a computer, touch screen **106** could be introduced in addition to the other user interface devices of the computer.

[0037] **FIG. 2** illustrates an illustrative embodiment of touch screen **106** in greater detail. Touch screen assembly **200** consists of touch sensor **202**, display **204**, and backlight **206** stacked or laminated together. Touch screens can be built in a variety of alternative ways as are well known in the art. For example, touch sensor **202** can be an active sensor employing capacitive, resistive, inductive, or other methods, or it can be a passive surface on which touch sensing is accomplished by optical, acoustic, or other methods. Capacitive touch sensors are ideally suited for use in the present invention due to their sensitivity, low cost, ruggedness, and suitability to small sensing areas. However, any touch screen technology would serve for the present invention.

[0038] Similarly, display **204** can be a liquid crystal display (LCD), organic light emitting diode (OLED) display, electroluminescent display, or any other type of small display suitable for mounting in a portable computer. LCD displays are ideally suited for use in the present invention due to their low cost and availability, but other types of displays may be employed. Display **204** may be color or monochrome, and need not have the same resolution, color capabilities, or other qualities as the main display of the computer.

[0039] The touch screen assembly may include a backlight **206** to enhance readability in all lighting conditions. In alternative embodiments, backlight **206** may be replaced by a frontlight, passive reflector, or other light source, or it may be omitted altogether.

[0040] Touch screen assembly **200** may include additional layers or components to assist the mounting or mechanical properties of the touch screen or to integrate the touch screen with other components of the computer system. The touch screen may also include hardened, antireflective, textured, or

other surface layers. The inclusion, omission, or nature of these additional layers and components is immaterial to the present invention.

[0041] Touch sensor **202** is connected to touch sensing controller **208**. The nature of controller **208** depends on the design of touch sensor **202** and its details are immaterial to the present invention. Likewise, display **204** is connected to a suitable display controller **210**, and backlight **206**, if present, is connected to backlight controller **212**. Each of controllers **208**, **210**, and **212** communicate with host computer **214**. In an illustrative embodiment, controllers **208**, **210**, and **212** are connected to a central touch screen controller **216** that connects to host computer **214** by a single interface **218**. Interface **218** may be a mouse interface such as PS/2, or a general purpose peripheral interface such as the Universal Serial Bus (USB). USB has the advantage of high bandwidth and wide availability. Any of controllers **208**, **210**, **212**, and **216** may be implemented as chips or discrete components, combined onto fewer chips or one chip, integrated with assembly **200**, or combined with other functions of host computer **214**. Host computer **214** may be embodied in the central processing unit of computer system **100**, a peripheral processor such as a USB host controller, or a combination thereof.

[0042] In an alternative illustrative embodiment, controllers **208**, **210**, and **212** may connect to host computer **214** through different interfaces. For example, touch screen controller **208** could connect as a conventional touch pad using a PS/2 interface, while display controller **210** and backlight controller **212** connect by USB or by a specialized display interface.

[0043] Because touch screen **106** of **FIG. 1** replaces a conventional touch pad, touch screen **106** usually serves as a conventional pointing device for the computer. For this reason, the touch screen must be able to interface to the computer as a conventional mouse. This is a further reason for interface **218** to be either a mouse interface such as PS/2, or a general interface such as USB that includes support for conventional mice. Interface **218** may also provide for an alternate or extended interface protocol that allows for additional information about finger activity to be communicated to computer **214**, and for computer **214** to control display **204** and backlight **206**. This additional finger activity information may include the absolute location of the finger on the sensor surface. When appropriate driver software is loaded onto computer **214**, the driver software can enable the alternate or extended interface protocol to support the user interface enhancements of the present invention. When other driver software, such as a conventional mouse or touch pad driver, is loaded instead, interface **218** can revert to mouse or touch pad compatibility using touch sensor **202** as a conventional touch pad, and controller **210** or **216** can operate the display autonomously, such as by furnishing a suitable default display image for display **204**.

[0044] When the touch screen is used as a conventional touch pad, finger motions on the touch sensor (e.g., in a cursor positioning region, which could identify a starting position) will typically cause corresponding motions of a cursor on the main display, and clicks of "mouse" buttons (or action control icons) **108** will typically cause special actions, such as selections on the main display. Tapping gestures may be interpreted as "mouse" clicks or other special actions, as disclosed in U.S. Pat. No. 5,543,591. Other gestures may also be recognized, such as scrolling motions as disclosed in U.S. Pat. No. 5,943,052. The default display image may include graphical icons to indicate special tapping or scrolling regions on the touch sensor surface or the default screen image may be a blank screen with only a manufacturer's logo.

[0045] In one embodiment, the cursor positioning region is denoted by the absence of icons for actions, other than cursor positioning. However, there are many different ways of identifying the cursor positioning region on the touch screen, such examples include, but are not limited to, a box could enclose the cursor positioning region, a shaded region or icon could cover the entire cursor positioning region, or an icon could be centered in an otherwise blank area, thus labeling the blank area as a cursor positioning region.

[0046] **FIG. 3** illustrates an example default image for use when the touch screen is operating as a conventional touch pad. **FIG. 3** depicts the image on the touch screen display as seen by the user. Image **300** includes arrow icons **302** and **304** indicating scrolling regions, an icon **306** indicating a corner tap region that simulates a right mouse button click, and an icon **308** which represents a logo for the computer vendor.

[0047] Alternatively, computer system **100** of **FIG. 1** can include a secondary pointing device, such as an isometric joystick located in keyboard **104** or an external mouse, which relieves touch screen **106** from the responsibility of functioning as primary pointing device in addition to its role as an enhanced user interface device.

[0048] A conventional touch pad with default screen image is just one of several general modes of usage that are envisioned for the touch screen of the present invention. Subsequent drawing figures illustrate several other usage modes that employ the touch screen as a fully interactive input/output device to enhance the user interface of the computer system. These general usage modes include "iconic,""auxiliary," and "pop-up" touch screen modes, each with a variety of possible applications. The same touch screen can operate in each of these various modes, or other modes, at different times. The different modes can also appear on the screen at the same time; for example, icons can appear in an auxiliary or pop-up image, or an auxiliary or pop-up image could be overlaid in a window on the iconic mode image instead of fully replacing that image.

[0049] **FIG. 4** illustrates an example of a first "iconic" usage mode of the touch screen. In the iconic mode, the screen displays an image that includes a number of small icons such as pictures or buttons. The touch sensor operates as a touch pad pointing device in iconic mode, in which finger motions and taps on the sensor are generally interpreted the same as when the touch screen operates as a conventional touch pad. The screen image in iconic mode may include elements in common with the default image of **FIG. 3**, as the two modes operate similarly. Iconic mode will generally display additional icons relating to software that is running on the computer and other aspects of the operation of the computer.

[0050] In the example image of **FIG. 4**, image **400** includes scroll arrow icons **402** and **404** and a touch region, such as illustrated by corner tap icon **406** in common with

FIG. 3. Logo **308** has been omitted from image **400** in this example to reduce clutter. In an alternate embodiment, non-critical graphics from the default image could be retained as a background image on which icons overlap; in yet another embodiment, a different image such as static or dynamic "wallpaper" may serve as a background image.

[**0051**] In example image **400**, additional icons have been added to represent various system status indicators and functions. Icon **410** defines a second touch region or a corner tapping region to activate the "back" function of web browsers and other software. As the user enables and disables special tap regions and changes their assigned functions, such as by using a software control panel, the tap region icons such as icons **406** and **410** can appear, disappear, move, and change in shape to reflect the current settings.

[**0052**] Icon **412** is a continuous display of the time and date. This icon would normally have no effect on the interpretation of finger taps within its region. Instead, a finger tap within its boundaries would be interpreted as a simulated mouse button click, just as if the tap occurred away from any icon. If every icon responded specially to finger taps, the main function of tapping to simulate a mouse click would become too inconvenient to use. A visual convention may be used to indicate which icons represent tap-sensitive regions; in the example of **FIG. 4**, dashed lines **426** and **428** are used to indicate these regions.

[**0053**] Icon group **414** includes the traditional set of status icons that appear on modem portable computers, such as numeric keypad lock, caps lock, scroll lock, hard disk activity, battery life, and system power. By locating these system icons on the touch screen display, the system designer eliminates the need for the special dedicated LED or LCD status displays that are typically used in prior art computers.

[**0054**] In some prior art portable computers, the dedicated system status displays are situated so that they are visible even when the cover of the computer is closed over the main display. The touch screen of the present invention could similarly be situated so that all or part of the screen image is visible when the cover is closed, for example, by causing the touch screen to protrude from under the cover or by cutting a notch in the cover over the location of the touch screen. This arrangement would allow the user to monitor battery recharging and other quiescent activities of the computer system while the computer is not in use.

[**0055**] Icon **416** is an e-mail notification status icon; icon **416** may, for example, change to a new shape or animated image to indicate that e-mail has arrived. Icon **418** similarly notifies the user of imminent appointments. These icons suggest a natural action that could be taken when the user taps on the icons, such as opening the associated e-mail reading or appointment scheduling software. Because these icons are located nearer the center of the touch sensing area and could easily be tapped by accident, icons **416** and **418** may be made sensitive to finger taps only when they have been activated by some separate means such as pressing a special function key on keyboard **104**.

[**0056**] Icons **420** and **422** represent commands to select pop-up applications on the touch screen. Icon **420** selects an application launcher. Icon **422** selects a calculator or numeric keypad. Like icons **416** and **418**, icons **420** and **422** may be made sensitive to finger taps only when the touch screen is in the activated state.

[**0057**] Icon **424** represents the volume control for the sound system and speakers of the computer. Icon **424** includes a visual slider and "thumb." The position of the thumb on the slider reflects the current volume setting. When the touch screen is in the activated state, finger motions within the volume control region can move the thumb to a different location on the slider to adjust the volume level. When the touch screen is not in the activated state, icon **424** is a visual display only and has no special interpretation when touched. Similar slider controls may be provided to adjust other system parameters such as the sound balance among several sound sources, the brightness and contrast of the main screen or touch screen, or the power management strategy.

[**0058**] The icons depicted in **FIG. 4** are illustrative of the types of icons that can be provided on the iconic mode screen. **FIG. 4** does not necessarily represent the ideal selection or placement of icons. Human-factors testing may be used to decide on the number, types, and placement of icons in the default iconic screen. Also, it may be advantageous to allow the user to select which icons are present and to rearrange the icons, possibly using a software control panel. Because the number of candidate icons likely exceeds available space, it may be desirable to provide multiple iconic screen layouts selectable by some means such as in the software control panel or by tapping on an icon on the touch screen.

[**0059**] It will be obvious to one skilled in the art that many other images, logos, status indicators, command buttons, controls, and other types of icons can share the touch screen display in the iconic usage mode. These icons can be purely display indicators, or they can indicate control regions that respond specially to finger motions and/or finger taps, either at all times or only when the touch screen has been activated in a special way. Some icons may be built-in by the system designer, such as the system status icons or the logo of the computer manufacturer. Other icons may be created and maintained by application software running on the computer, such as an e-mail notification icon.

[**0060**] The activated state of the touch screen may be indicated by a visual convention. **FIG. 5** illustrates the touch screen image of **FIG. 4** modified to indicate the activated state of the touch screen using a dashed line around each icon that is touch-sensitive in the activated state. In image **500**, dashed lines **516**, **518**, **520**, and **522** have surrounded certain icons to indicate that finger taps in the regions near these icons will be interpreted as special commands to the icons. Similarly, dashed outline **524** indicates that finger motions in the volume control region will adjust the setting of the control. Outline **512** for the time and date icon has become dashed to indicate that a tap on this icon will activate a special function such as setting the time or accessing a world clock. Outline **514** for the system status icons remains solid to indicate that, in the example of **FIG. 5**, these icons have no special tapping functions in the activated state. Dashed lines **526** and **528** remain to indicate that the corner tap regions continue to have their special tap interpretations when the touch screen is in the activated state. Many other visual conventions would serve equally

well to indicate touch-sensitive icons, such as solid or colored lines, colored or inverted backgrounds, changes in brightness or coloration of the activated icons, changes in shape or animation of the activated icons, or other well-known conventions for highlighting a portion of an image.

[0061] The example of **FIGS. 4 and 5** illustrates the same set of icons in the activated and unactivated state. However, activation of the touch screen could also create additional icons that are not present, for example to reduce clutter, when the touch screen is not in the activated state. Existing icons could also be removed or rearranged, although to avoid confusion, this could be done only to replace icons not useful in the activated state, such as icon **414** of **FIG. 4**, with other icons that are most useful when activated, such as icons similar to icons **420** and **422**.

[0062] There are many possible alternative mechanisms for the user to signal the activation of touch screen icons. In the simplest case, the icons are either always inactive or always active for tapping or motion commands. The corner tapping and scrolling region icons **302**, **304**, and **306** of **FIG. 3** and **402**, **404**, **406**, and **410** of **FIG. 4** are examples of icons that are active at all times. The logo icon **308** of **FIG. 3** and system status icons **414** of **FIG. 4** are examples of icons that are inactive at all times. For simple touch screen images like that of **FIG. 3**, all icons may fall into these simple categories and no overt activation mechanism is needed. For more elaborate touch screen images like that of **FIGS. 4 and 5**, an overt activation mechanism is necessary for those icons that must respond to touch but cannot reasonably be made touch sensitive at all times.

[0063] In an illustrative embodiment, a key on the main keyboard **104** of computer system **100** of **FIG. 1** is designated as the touch screen activation key. **FIG. 6A** illustrates a portion **600** of the keyboard featuring several keys. Most keys of the keyboard, such as letter "Z" key **602**, have preassigned functions that do not overlap well with touch screen activation. Even the existing shifting keys such as shift key **604**, control key **606**, and Alt key **614** are not suitable because they are often pressed in conjunction with mouse clicks in application software for features such as extending selections; hence, it is desirable for the user to be able to tap normally on the touch sensor to simulate a mouse click while these shifting keys are pressed.

[0064] Function or "Fn" key **608** is common on the keyboards of portable computers. This key, when held down, changes the interpretations of various other keys to perform special control functions. For example, in one portable computer, the arrow keys change to screen brightness controls, certain letter keys change to become a numeric keypad, and various other keys change to control the external video and various other functions. The alternate "Fn" functions of the various keys are often indicated by blue writing next to the white writing indicating the primary function of a key. Because the "Fn" key is often absent on desktop computers, software typically does not give special interpretations to mouse clicks in conjunction with the "Fn" key. The usage and functionality of "Fn" key **608** coincides well with the function of activating the touch screen. In one illustrative embodiment, holding down the "Fn" key causes various icons on the touch screen to be activated with visual feedback as shown in **FIG. 5**, in addition to the normal action of redefining various keys of the main keyboard. Releasing the "Fn" key causes the touch screen to revert to its pointing device usage at the same time as the keys of the main keyboard revert to their primary functions.

[0065] If "Fn" key functions are indicated by a color code (such as blue writing), this color code can be employed on a color touch screen for extra mnemonic effect. For example, blue outlines or coloration can be used on the icon itself or in a background or outline to indicate those icons whose behavior will change when the touch screen is in the activated state. The outline or background could then change from blue to white when the touch screen is activated, signifying that the icons are now sensitive to touch.

[0066] Computers intended for use with the Microsoft Windows® operating system often include a "Windows" key **610**. The "Windows" key also changes the interpretations of various other keys on the computer keyboard while it is held down. The "Windows" key is another candidate for a touch screen activation key with semantics similar to those disclosed for the "Fn" key. Those practiced in the art will recognize that certain other keys that appear on some portable computer keyboards, such as the "AltGr" key, may also be suitable candidates for a touch screen activation key.

[0067] In an alternate embodiment, a new key **612** can be added on or near the keyboard to serve as a dedicated touch screen activation key. Key **612** could operate as an activation shift key for which the touch screen is activated for the duration that the key is held down. Or, key **612** could operate as an activation prefix key for which the touch screen is activated after the key is struck and until an icon is tapped. In yet another embodiment, key **612** could operate as a toggle key that alternately activates and deactivates the touch screen each time it is struck. Any of these schemes or others would work, but it may be advantageous to use an existing key such as "Fn" key **608** or "Windows" key **610** instead of a dedicated key **612**. Using an existing key simplifies keyboard design and is more familiar to users accustomed to standard keyboards. However, it may be advantageous to label the existing key with an icon or lettering to indicate its dual function as a touch screen activation key in addition to its normal label, as illustrated by key **616** of **FIG. 6A**.

[0068] Many other touch screen activation mechanisms are possible alternatives to a keyboard key. In one embodiment, an additional mouse button is provided adjacent to buttons **108** of **FIG. 1** to activate the touch screen. Alternatively, a special touch sensitive region or second touch sensor could be provided that activates the touch screen when touched. **FIG. 6B** illustrates one possible arrangement of such a button or touch sensor. Toroidal button or touch sensor **632** surrounds all or part of the touch screen **630**. Toroidal button or sensor **632** is distinct from conventional "mouse" buttons **634** and **636**. In one usage, the toroidal button would activate the touch screen when touched or pressed. Alternatively, the touch screen icons could remain active except when toroidal button or sensor **632** is touched or pressed. This latter usage may be advantageous since the user can be expected to keep the hand near the keyboard or near conventional buttons **634** and **636**, and therefore also near sensor **632**, during conventional operation of the computer when touch screen **630** is mostly likely to be operated unconsciously as a pointing device.

[0069] Another possible activation mechanism is to provide a region on the touch screen which is always active, and

in which finger taps are interpreted as a signal to enter or toggle the activated state of the touch screen. A software control panel could offer the activation function as one of the possible functional assignments of corner tap regions **406** and **410** of **FIG. 4**.

[**0070**] Yet another mechanism is for the user to click on a soft button or icon on the main display to activate the touch screen. Numerous other activation mechanisms are well known that could serve for touch screen activation, such as finger motion gestures, voice commands, foot switches, retinal gaze tracking, etc. Software applications that make use of the touch screen can offer additional, application-specific activation mechanisms.

[**0071**] In yet another embodiment, icons are individually activated by being touched in a special way instead of by an overall touch screen activation state. For example, single taps near an icon could be interpreted as normal mouse clicks but rapid double taps could trigger the "activated" function of the icon. Alternatively, touching an icon with multiple fingers, or hovering the finger over an icon without touching the surface of the touch screen, or holding the finger steady over an icon for a given duration, could trigger the activated function of the icon.

[**0072**] Some touch screen technologies are sensitive to other objects, such as a pen, pencil, or pointer, in addition to fingers. In such devices, a finger tap could trigger an activated function while a pen tap would be interpreted as a normal mouse click, or vice versa. Or, a special button could be provided on the body of the pen that triggers the activated function of an icon when pressed.

[**0073**] It is also possible to provide several of these alternate mechanisms at once. These multiple activation mechanisms could be synonyms in that they all activate the same special function of each icon, or different activation mechanisms could activate different special functions of the icons. Multiple different special functions should be used with caution because of the likelihood of confusing the user.

[**0074**] With iconic screen images such as that of **FIGS. 4 and 5**, it is desirable to include many small icons on the screen to provide access to a maximum number of features. As shown in **FIG. 7**A, such small icons **702** may be smaller than finger **700** and may be completely obscured by the finger when the finger touches them. Because the finger will cover the icon only momentarily, this effect may not be a serious problem. However, various techniques can be employed to solve the problem of obscuring small icons, and in an illustrative embodiment the screen images are designed so that the icons are either large enough to avoid being obscured, or situated so that the user can operate them even when they are momentarily obscured, or provided with a mechanism to eliminate the problem of obscuring small icons.

[**0075**] FIGS.7B-7E illustrate several such mechanisms. In the mechanism of **FIG. 7**B, icon **710** expands whenever finger **700** passes over it. In the mechanism of **FIG. 7**C, an image **722** of the icon or image area under the finger is displayed in "callout"**720** adjacent to finger **700** or elsewhere on the screen. In the mechanism of **FIG. 7**D, finger **700** selects not the icon directly under the finger, but the icon **730** under a "hot spot"**732** displaced enough from the center of finger contact to be visible around the finger. As shown in **FIG. 7D, a** crosshair may help to visually indicate the hot spot **732** to avoid confusion. The mechanism of **FIG. 7E** uses the property that certain touch sensing technologies, such as that disclosed in U.S. Pat. No. 5,543,591, compute the centroid of all finger contact on the sensor. With such sensors, the user can select icon **744** without obscuring it from view by placing two fingers **740** and **742** on either side of the icon instead of a single finger directly on the icon. Crosshair **746** may be provided to make the centroid of finger contact more visually apparent.

[**0076**] **FIG. 8A** illustrates an example of a second "auxiliary" usage mode of the touch screen of the present invention. In the auxiliary mode, the touch screen displays an auxiliary image specific to a software application that is running on the computer. In an illustrative embodiment, a software application displays its auxiliary image only when it has the "input focus" as determined by the operating system. In most computer operating systems, application windows on the main display screen are given the focus based on which was last clicked by the pointing device, or on which currently contains the cursor. The auxiliary image for an application may include graphic icons and buttons that may or may not coincide with those of the iconic mode. Alternatively, the auxiliary image may be a pure image, such as an advertisement or a set of notes accompanying a presentation.

[**0077**] In the auxiliary mode, finger motions and/or finger taps would typically be given a special interpretation by the application. If the application treats finger motions specially, the touch screen will be unable to move the cursor on the main display as long as the application imposes its special interpretation on finger motions. This may be acceptable if an alternate cursor motion device or mechanism is present, or if the application does not need a cursor, or if the special interpretation lasts for only a brief duration. Alternatively, if the application treats only finger taps specially, then the user can use the touch screen to move the cursor on the main display, but the user must use the "mouse" buttons **108** of **FIG. 1** to click or select items on the main display. In another alternative, the application may display an auxiliary image but allow the touch screen to interpret finger motions and taps in the same way as the iconic mode. In this latter alternative, if the auxiliary image includes buttons or control icons, then a special activation mechanism must be used to activate the buttons or controls as disclosed for the iconic mode. Applications may divide the screen into regions or icon image areas that interpret finger motions or taps in different ways, analogous to the special treatment of taps in corner regions **406** and **410** and the special treatment of finger motions in scrolling regions **402** and **404** of **FIG. 4**. In an illustrative embodiment, each application may choose any of these alternatives, or other alternatives, for its auxiliary screen as best fits the needs of the application.

[**0078**] In the example of **FIG. 8**A, the touch screen illustrates an auxiliary image for a slide presentation. Slide presentation software, such as Microsoft PowerPoint®, typically uses the entire main display of the computer in full-screen mode to display the current slide. Because the main display may be shown to an audience or linked to a video projector, the main display must show only the slide image itself. The touch screen displays an auxiliary image **800** with information useful to the presenter. Region **802** displays the slide number, title, and speaker's notes. Region

804 displays the title or preview image of the next slide, and region **806** similarly displays the previous slide in the presentation. Regions **804** and **806** are finger-tappable buttons to advance the presentation forward or backward by one slide. Region **802** is configured so that a finger tap brings up a menu of additional presentation options; in one example presentation software system, tapping on region **802** would simulate a right mouse button click. The slide presentation software would be configured to display auxiliary image **800** only during a full-screen presentation. At other times, the software would allow the touch screen to revert to iconic mode with the touch sensor serving its usual role as a pointing device to operate the software.

[**0079**] Those skilled in the art will recognize that the slide presentation application of **FIG. 8A** is representative of a class of applications that can benefit from leaving the entire main display free to display dedicated images. Another example is a software player for DVD movies or videos. DVD players usually include controls such as pause, reverse, fast forward, and chapter select. However, it would be distracting to place these control icons on the main display of the computer when a movie is playing. In the example of **FIG. 8**B, the DVD player places a small control panel **820** on the touch screen, reserving the entire main display for movie viewing. Control panel **820** includes status icons **822** displaying track information and timing, buttons **824** for operations such as stop and fast forward, and volume control **826**. During full-screen movie viewing, buttons **824** and control **826** would respond to touch to control the playing of the movie. When the DVD viewing software is not in full-screen mode, the touch screen could be allowed to revert to normal iconic mode, or control panel **820** could remain on the touch screen display but with buttons **824** and controls **826** active only when the touch screen is in the activated state.

[**0080**] Similarly, many computer systems can play audio from music CD's. Users typically listen to CD's as background music while doing unrelated work on the computer. CD playing software typically displays a control window very similar to that of the DVD software. This window can obstruct the view of the application running on the main display, and would usually be moved to a touch screen display very similar to that of **FIG. 8B**.

[**0081**] **FIG. 8C** illustrates another application involving Internet web browsers. Web pages often include advertisements along with the main information of the web page. Some browsers and services offer to filter out the advertisements to reduce visual clutter, but such services encounter great resistance from web providers who depend on advertising revenues. Instead, the browser or service could move the advertisement image onto the touch screen where it remains plainly visible but less obstructive to the main web page. In addition, the touch sensor system could employ a validation mechanism using any of numerous well known digital signature means to allow the display of only those images which the user has allowed or for which the advertiser has paid a licensing fee. In **FIG. 8**C, image **840** includes advertisement image **842** drawn from an unrelated web page displayed on the main display. In this application, the touch sensor would normally operate as a pointing device, but when the touch screen is in the activated state, tapping on image **842** would instead be interpreted as a click on the advertisement itself.

[**0082**] **FIG. 8D** illustrates yet another potential application involving word processors, such as Microsoft® Word, and document viewers, such as Adobe Acrobat®. These software tools often display auxiliary information such as a table of contents or a set of thumbnail page images to provide context along with the main page or pages on display. This auxiliary information adds clutter and takes up space that could otherwise be devoted to page viewing. In the example of **FIG. 8**D, auxiliary information **862** has been moved to touch screen **860**, leaving more room on the main display for page viewing. Corner tap regions **866** and **868** have been retained but their functions have changed to functions better suited to the document viewing application; region **866** selects the previous page and region **868** selects the next page. Scrolling region **870** has been retained from the default iconic screen, as scrolling is an important function of a document viewer. When the touch screen is in the activated state, tapping on any of thumbnails **864** would cause the page viewer to display the selected page, and scrolling region **870** scrolls thumbnails **864** within area **862** instead of scrolling the document view on the main display.

[**0083**] Another class of applications that can benefit from the touch screen in auxiliary mode is those applications that can benefit from displaying additional or subsidiary information. Many computer operating systems and software applications today provide pop-up help that appears automatically on the screen when the cursor is held still on an icon or button. The pop-up help displays a brief explanation of the icon or button, allowing the user to know ahead of time what will happen if the icon or button is clicked. Pop-up help is usually restricted to brief one-line descriptions, as larger automatic pop-up help windows would obstruct the display. When large unsolicited on-screen help displays have been attempted, as in Microsoft's animated paperclip assistant, users have often found the help feature to be more distracting and obstructive than useful.

[**0084**] According to the present invention, applications can display more extensive pop-up help or other explanatory or subsidiary information on the touch screen when the cursor covers an icon or button on the main display. Because touch screen help text does not obscure anything on the main display, it can be quite extensive, and it can appear immediately without waiting for the cursor to hold still over the icon for a period of time. Touch screen help can also be offered for user interface elements that normally are not well suited to pop-up help for visual design reasons, such as the selections within pull-down menus.

[**0085**] **FIG. 9** illustrates an example of subsidiary help text on the touch screen of the present invention. When the cursor covers a user interface element on the main display for which help is available, the normal iconic or auxiliary screen image is replaced by a new auxiliary image **900** that persists as long as the cursor remains on the element on the main display. Image **900** includes help text **902** describing the object, in this case the "Format Painter" toolbar icon or menu item of a document preparation tool. Because the user will not necessarily notice that the standard iconic touch screen image has been replaced, permanently active touch regions such as corner tap regions **904** and **906** should be carried over from the replaced image. The rest of the touch screen image is free for help text or other subsidiary information. In the example of **FIG. 9, a** button **908** is also provided that can be tapped to obtain more help. Again,

because the user may not be aware that such buttons have appeared, button **908** should usually be sensitive to finger taps only when the touch screen is in the activated state.

[**0086**] Some software applications already include detailed help text for many user interface elements. This help text may be intended for display when the user invokes an explicit context-sensitive help command for the element. In the present invention, this pre-existing detailed help text can be adapted for display on the touch screen as well, possibly with little or no modification to the application software itself.

[**0087**] All of the preceding examples have featured auxiliary screens tied to a particular application. It is also possible for a particular window or dialog within an application to have an associated auxiliary screen. For example, the Open File command in most applications brings up a characteristic dialog window on the main display. This dialog includes a list of files and directories, a space for typing in a file name, and various buttons for navigating the file system. In many cases, the software application calls on the underlying operating system to supply a standardized dialog for choosing a file. An application, or the operating system itself, could supply an auxiliary screen image with additional buttons, controls, or displays to help the user select a file.

[**0088**] Some dialogs must interact with the main display image of an application. For example, the text find and replace dialog of a word processor typically must stay open as the user calls for repeated searches and replacements in the document, but the dialog tends to get in the way of the view of the document being searched. Word processors employ elaborate heuristics to try to keep the dialog box and the focus of attention within the document out of each others' way.

[**0089**] **FIG. 10A** illustrates a better solution employing the touch screen of the present invention. The find/replace dialog **1002** is displayed on touch screen **1000**, leaving the main display free to display the document unobstructed. To aid user understanding, dialog **1002** is designed to resemble a conventional dialog box with title bar **1004**, text entry areas **1006** and **1008**, functional buttons **1010** and **1012**, and close button **1014**. However, some of the conventional dialog elements have been adapted to best suit the touch screen interface. Functional buttons **1010** and **1012** are located in the corners of the touch screen surface so that they can be made active even when the touch screen is not in the overall activated state. If buttons **1010** and **1012** were drawn in the conventional way, similar to button **908** of **FIG. 9**, then it would be too confusing to the user for buttons **1010** and **1012** to be sensitive to taps except when the touch screen is in the activated state. Similarly, close button **1014** is located near the corner of screen **1000** so that it can safely be made active at all times. Text entry areas **1006** and **1008** would be filled in by the user at the beginning of the search operation, and would then normally be inactive; tapping on them when the touch screen is in the activated state could allow the search or replace text to be changed. Because text entry on a dialog box is easier to understand on the main display, it may be advantageous for dialog **1002** to appear on the main display during entry of text into areas **1006** and **1008**, and then to move to touch screen **1000** during the repeated search operation. In addition or alternatively, a user command, such as a gesture could be provided to move any dialog between the main display and the touch screen at the user's discretion.

[**0090**] Similarly, many applications display "alert" dialogs, typically appearing at unexpected times with brief text messages, to alert the user of errors or other irregular events. Alert dialogs can confusingly obstruct the view of the very operation that caused the alert, and are another good candidate for moving to the touch screen. Applications often call on standard operating system services to display alert dialogs, so the task of moving alerts to the touch screen can be accomplished in the operating system without the cooperation of individual software applications.

[**0091**] **FIG. 10B** illustrates yet another application of a touch screen. Many computer games use the mouse as a game controller device in lieu of joysticks or other specialized game controller hardware. A touch pad emulating a mouse serves as a passable game controller, but often the touch pad can be made into a superior game controller by adjusting its behavior to best fit a particular game. Experiments with touch pads have shown that a touch pad reprogrammed in this way can be an excellent game controller, equaling or exceeding the performance of some dedicated game controllers. However, with conventional touch pads it has been too confusing to invisibly redefine the behavior of the touch pad for each game. As shown in **FIG. 10**B, the touch screen of the present invention solves this problem by displaying the control layout established by the game. In this example, a flight simulator displays an image **1030** including regions **1032** and **1034** similar to conventional scroll regions to control the throttle and flaps, and tap regions **1036** and **1038** to control the landing gear and change the view presented on the main display. Each of these controls is clearly marked by text or symbols on the touch screen to help the user learn the controls.

[**0092**] To be effective game controls, regions **1032**, **1034**, **1036**, and **1038** must be sensitive to touch at all times, without requiring the touch screen to be in an activated state. The remaining area of screen **1030** may be used for normal cursor motion. If screen **1030** includes many game controls, there may be insufficient area remaining to support cursor motion. Depending on the game being controlled, it may or may not be acceptable to omit the cursor motion function. If cursor motion is required, one solution is to invert the sense of activation so that the touch screen operates as a normal pointing device only when it is in the activated state. Another solution is to provide a small cursor control region, such as region **1040**, that operates on different principles from a regular touch pad. Region **1040** could serve as a relative cursor motion device, where placing the finger in the region and then rocking the finger a small distance in any direction causes steady cursor motion in the indicated direction. These or similar mechanisms could be used in any auxiliary or pop-up screen that must support cursor motion despite using most of the screen area for other functions.

[**0093**] Many software applications provide drop-down menus or toolbars on the main display to invoke various operations and commands of the application. Another beneficial use of the touch screen of the present invention is to move or duplicate some or all of these menus and toolbars onto the touch screen. **FIG. 10C** illustrates an example in which image **1060** includes icons **1062** drawn from a typical

toolbar. By activating the touch screen and tapping any of these icons, the user can invoke the corresponding function in the software application. Because these icons would appear in the same relative location on the touch screen every time the application is used, the user can learn their locations by feel and thus avoid the distracting task of moving the cursor away from the natural focus of attention and onto the menu or toolbar. Displaying toolbar icons **1062** on the touch screen allows the user to locate the icons in the learning phase, before the locations of the icons are known by feel.

[**0094**] Those practiced in the art will see that many other types of applications can make use of auxiliary displays and controls on the touch screen. For example, spelling and grammar checking software could display lists of correction choices without obstructing the text being reviewed. The set of examples disclosed and illustrated here in no way limits the scope of applications that can benefit from an auxiliary touch screen according to the present invention.

[**0095**] **FIG. 11** illustrates an example of a third "pop-up" general usage mode of the touch screen of the present invention. In the pop-up mode, the touch screen displays a special image much as in the auxiliary mode. The pop-up mode allows all the same display elements on the touch screen and all the same alternative interpretations of finger actions on the touch sensor as in the auxiliary mode. However, the pop-up image appears in response to a user command or other event in the host computer and is not associated with any particular software application on the main display.

[**0096**] In the example of **FIG. 11**, the pop-up image is an application launcher. When the application launcher is invoked, image **1100** replaces the previous image on the touch screen. Image **1100** includes various icons **1102** representing commonly used tools and software applications on the computer. The set of applications shown may be predetermined or may be chosen by the user. When the user taps the finger on one of icons **1102**, image **1100** disappears and is replaced by the original touch screen image, and the selected application software is launched. Typically, this application would be a conventional software application such as a word processor running on the main display of the computer, but some of icons **1102** may represent system commands (such as shutting down the computer), other tools (such as another pop-up application on the touch screen), or links to additional application launcher screens. The user can also tap on icon **1104** to exit the application launcher screen without invoking any application.

[**0097**] Pop-up screens such as the application launcher of **FIG. 11** may be invoked by any of various well-known means for invoking applications, such as a keyboard key, an icon like icon **420** or comer tap region like region **410** of **FIG. 4**, or the "Start" menu of Microsoft Windows®.

[**0098**] Pop-up screens may be implemented as regular applications as viewed by the operating system; in this case, the application would not create a visible window on the main display, but it would create a touch screen image using the same mechanisms that other applications would use to create an auxiliary touch screen image. In an alternate embodiment, pop-up screens like that of **FIG. 11** could be implemented specially within the touch screen driver software, or they could be implemented in the touch screen controller hardware such as controller **216** of **FIG. 2**.

[**0099**] **FIG. 12** illustrates a pop-up calculator application that operates entirely within the touch screen. Image **1200** includes the familiar numeric display **1202** and a matrix of buttons **1204** of a calculator. The user taps on the button icons to operate the calculator in the usual fashion. The user taps on button **1206** to close the calculator and restore the touch screen to its previous image. The calculator operates autonomously with respect to the applications visible on the main display of the computer. This autonomous behavior is particularly valuable when the calculator is being used in tandem with an application on the main display, such as a database application looking up numeric data. In the example of **FIG. 12**, buttons **1208** and **1210** are provided to allow numbers to be pasted back and forth between the calculator and the active application on the main display.

[**0100**] Computer keyboards traditionally include a numeric keypad, but portable computer keyboards rarely have room for a conventional keypad. Portable computer system designers are forced to adopt awkward solutions such as the "Fn" key. A pop-up numeric keypad screen very similar to the calculator of **FIG. 12** could serve the role of the numeric keypad in a portable computer. This keypad screen could be invoked by the "NumLock" key already provided on computer keyboards for activating the numeric keypad.

[**0101**] Many computer operating systems provide a magnification tool to assist the visually impaired. This tool typically creates a window on the main screen that displays a magnified copy of the display image surrounding the cursor. This magnifier window can obstruct useful information on the main display. According to the present invention, as illustrated in **FIG. 13**A, the magnifier can instead take the form of a pop-up image **1302** on touch screen **1300**, leaving the main display undisturbed. Unlike the examples of **FIGS. 11 and 12**, the magnifier pop-up would probably be left displayed much of the time that the computer is used. This pop-up application would therefore leave the touch sensor operating as a conventional pointing device; hence, corner tap regions **1304** and **1306** are retained. When the touch screen is in the activated state, the magnifier application can offer additional accessibility features on the touch screen. In the example of **FIG. 13**B, in the activated state, touch screen **1320** replaces image **1302** with an image of controls such as magnification level adjustment **1322**. Also, close box **1324** appears in the activated state to allow the user to turn off the magnification feature. In an alternate embodiment, in the activated state, the magnifier activates features to assist operation of small on-screen controls.

[**0102**] In an alternative magnification mode, the main display image is reduced and moved to the touch screen display, and then a magnified view of a portion of the image is shown on the main display. This has the advantage that the main display is larger and likely to have greater clarity and color depth than the touch screen, and will thus be a better detail viewing device for the visually impaired.

[**0103**] Debugging is a task that greatly benefits from a secondary display. Computer programmers today sometimes attach a second display monitor to their computers so that the program under debugging can operate undisturbed on the first display monitor. These second displays are costly and inconvenient, particularly on portable computers. As shown in **FIG. 13C, a** debugger could be implemented instead as

a pop-up application on the touch screen of the present invention, providing the benefits of a secondary debugging display with no extra cost or bulk. In the example of **FIG. 13**C, image **1340** includes command buttons **1342** and source code display window **1344**.

[**0104**] Users of ideographic languages like Chinese and Japanese typically rely on input methods beyond the simple direct keystroke mapping used in Western languages. A variety of input methods are in use for ideographic languages, many of which require or benefit greatly from providing visual feedback to the user through a special window. This window can obstruct the application for which the input is intended. According to the present invention, the input method dialog can be implemented as a pop-up image on the touch screen. One popular input method is handwriting recognition, in which case the touch screen can also serve as the handwriting input device for added benefit.

[**0105**] **FIG. 14** illustrates an example Chinese handwriting entry system on touch screen **1400**. Handwriting entry area **1402** responds to finger touch to enter a Chinese character. In this application, the touch screen sensing technology advantageously senses pens as well as fingers; although handwriting with fingers has been shown to work quite well, many users prefer to write with a pen. Pen or finger motions in area **1402** can leave an "ink" trail **1408** on the touch screen display to allow the user to see the character as it is being drawn. Once a character is drawn in area **1402**, the software attempts to recognize it as a valid Chinese character. The software creates an ordered list of possible matches, which are displayed in area **1404**. The user can touch one of the match characters in area **1404** to "type" the selected character into the application running on the main display. Area **1406** contains touch-sensitive buttons to control the character recognition software in various ways.

[**0106**] Handwriting with "inking" is also useful in applications such as signature capture, signature recognition, and sketching, all of which are enhanced by the touch screen of the present invention.

[**0107**] Another useful class of pop-up screen applications is in the area of security. Portable computers are especially vulnerable to theft, so many portable computers include some kind of password or biometric interlock. For maximum effectiveness, the interlock should validate the user's identity before the main processor of the computer is even allowed to run. Because the main display is operated by the main processor of the computer, the security interlock would need to use alternate output mechanisms to interact with the user. The touch screen of the present invention provides an excellent user interface device for a security interlock. The software that manages the interlock can be implemented in the touch screen controller itself, or in another peripheral controller within the computer. This implementation fits well with the architecture of many portable computers today, where a peripheral controller is already present in between the main processor and the touch pad, and this peripheral controller is also already tasked with power management and system reset control for the main processor.

[**0108**] **FIG. 15A** illustrates a pop-up screen **1500** that appears when the computer system is first switched on. The user must enter a correct personal identification number (PIN) on keypad icons **1502** before the main computer processor will operate. In an alternate embodiment, the user enters a signature on the touch screen or uses some other mechanism such as a smart card or fingerprint to authenticate himself or herself to the system.

[**0109**] **FIG. 15B** illustrates an exemplary hardware architecture implementing the security interlock of **FIG. 15A**. Computer system **1520** includes touch screen module **1522**, which in turn contains the hardware and control circuitry illustrated in **FIG. 2**. Touch screen **1522** communicates to peripheral controller **1524**. Controller **1524** also manages other peripherals **1526** such as keyboards, external pointing devices, and optional biometric authentication devices. During operation of the computer, controller **1524** serves as a conduit between touch screen **1522** and central processor **1528**. Central processor **1528** in turn operates other devices **1530** such as the main display and hard drive. Power supply **1532** powers central processor **1528** as well as all other components of the system. At system start-up, power supply **1532** withholds power from processor **1528** until it receives a signal from controller **1524** by direct connection **1534** stating that the user has been authenticated and system start-up can proceed. Alternatively, controller **1524** holds processor **1528** in reset, or it simply withholds access to the keyboard, touch sensor, and other user interface peripherals, hence rendering the computer system useless until the user is authenticated. In yet another alternative, controller **1524** could participate in higherlevel security functions such as delivering a decryption key for data stored on a hard disk.

[**0110**] The security interlock of **FIG. 15**A and the debugging screen disclosed previously are examples of the general class of applications that use the touch screen to communicate with the user when the rest of the computer system is indisposed due to special circumstances. Another example of this class would be the reporting of information about hardware failures in vital system devices such as the keyboard and the hardware of the main display.

[**0111**] Many other applications of pop-up screens are supported by the touch screen of the present invention. For example, pop-up games could be implemented entirely on the touch screen, leaving the main display unobstructed.

[**0112**] Referring back to **FIG. 2**, touch screen assembly **200** may advantageously include a backlight **206** or an equivalent. Backlights draw more power than the other components that make up a touch screen, so it is advantageous to switch the backlight off when it is not needed. In an illustrative embodiment, backlight controller **212** is capable of dimming or extinguishing the backlight at the request of controller **216** or host computer **214**. Controller **216** and host computer **214** may use heuristics to switch the backlight on and off without explicit direction by the user. For example, the backlight could be switched on if an application installs an auxiliary screen image that replaces the default iconic image, and then switched off if the touch screen goes a certain amount of time without being used. Similarly, the backlight could be switched on whenever the touch screen is in the activated state.

[**0113**] Switching on the backlight when the touch screen is activated has the added benefit of reminding the user that the behavior of the touch screen has changed. The backlight can serve more generally as an attention mechanism for software applications and for the operating system. For example, the backlight can be flashed on or off to notify the user of the arrival of new e-mail or of an impending

appointment. Many computer operating systems use sounds to alert the user of a variety of errors and events, but portable computers are sometimes used in public places where the sound must be turned off. The backlight can serve as a replacement alert in this situation. This feature is especially useful when alert dialogs are moved onto the touch screen from the main screen as disclosed in relation to **FIG. 10A**. Alert dialogs obstruct the view of the application data or interaction that may have raised the alert; by moving the alert to the touch screen and calling the user's attention to it by flashing the backlight, the present invention can improve the effectiveness of alert dialogs throughout the operating system.

[**0114**] If display **204** is a color display, then the system can flash backlight **206**, or color display **204** itself, in different colors to signal different types of alerts to the user. In this alternative, the use of color is analogous to the use of different sounds for audible alerts, and the touch screen may implement a mapping from standard sounds supplied by the operating system to standard color alerts.

[**0115**] Conversely, in some systems backlight **206** may be omitted to save space, power, or cost. On such systems, an alternate attention mechanism may be provided to alert the user when the touch screen is activated or changed to a new image with different active buttons. Suitable attention mechanisms include audible alerts, an icon or special cursor shape on the main display of the computer, an LED mounted near the touch screen, or a tactile feedback mechanism integrated with the touch screen.

[**0116**] The touch screen of the present invention must provide a mechanism for application software running on touch screen controller **216** or host computer **214** to create icons such as those shown on the iconic screen of **FIG. 4** and to create auxiliary and pop-up images such as those shown in FIGS. **8-15**. Various mechanisms are possible to accomplish this.

[**0117**] If the software that manages an icon or pop-up screen resides in touch screen controller **216**, then the software has direct access to touch sensor **202** and display **204** via controllers **208** and **210**. The software on controller **216** can interpose its own images into the sequence of images it receives from host **214** for display. The software on controller **216** can also intercept finger touch information from sensor **202** before sending this information to host **214**. By these means, icons and pop-up screens can be implemented by software entirely in controller **216** with no participation by host **214**. Depending on the nature of interface **218**, controller **216** may also be able to send keystroke information to host **214** to allow its icons and pop-up screens to control host **214** by simulated keystrokes.

[**0118**] In an illustrative embodiment, many icons, auxiliary screens, and pop-up screens are implemented by various software applications running on host **214**. To coordinate access to the touch screen by these various applications, host **214** includes driver software that serves as a conduit between software applications and touch screen controller **216**.

[**0119**] **FIG. 16** illustrates an exemplary software architecture for the touch screen of the present invention. Touch screen architecture **1600** consists of hardware layer **1602**, driver layer **1604**, and application layer **1606**. Those skilled in the art will recognize that many other software architectures are equally able to implement the user interface enhancements disclosed herein.

[**0120**] Hardware layer **1602** includes touch screen module **1610**, which in turn includes touch screen controller **216** of **FIG. 2**. Touch screen module **1610** connects to peripheral controller **1612**, which is included in host computer **214** of **FIG. 2**. Peripheral controller **1612** would be a USB host controller subsystem in the case that the USB protocol is used. Peripheral controller **1612** is operated by hardware driver **1614**. Hardware driver **1614** is supplied by the operating system of the computer and is not particular to the present invention.

[**0121**] Driver layer **1604** includes touch screen driver **1620**, which communicates with hardware driver **1614** to operate the touch screen hardware. Touch screen driver **1620** communicates with pointing device driver **1622**. Pointing device driver **1622** is supplied by the operating system and is responsible for operating mice and other pointing devices. When the touch sensor is operating as a conventional pointing device, touch screen driver **1620** converts sequences of finger positions reported by touch screen **1610** into motion signals similar to those produced by a mouse. Touch screen driver **1620** also examines the finger presence indication from touch screen **1610** to recognize finger tapping gestures. U.S. Pat. No. 5,543,591 discloses methods for computing tapping gestures on a touch pad sensor. These motion and gesture signals are conveyed to pointing device driver **1622** to cause cursor motion and clicking compatible with a mouse or conventional touch pad.

[**0122**] Touch screen driver **1620** also operates application programming interface (API) layer **1624**. Software applications running on the computer, represented in **FIG. 16** by software applications **1640**, **1642**, and **1644** in application layer **1606**, can use API **1624** to obtain special access to the touch screen. API **1624** exports a variety of touch pad and touch screen commands to the applications in application layer **1606**. These commands include requests for information about finger and "mouse" button activities on the touch sensor, as well as requests to override the cursor motion normally conveyed to pointing device driver **1622** with different cursor motion generated by the application based on finger movements. The API commands also include requests to display or update an icon on the iconic screen image, or to display or update a full-screen auxiliary or pop-up image.

[**0123**] Touch screen driver **1620** is responsible for deciding among conflicting API requests. For example, touch screen driver **1620** may consult pointing device driver **1622** or other operating system components to determine at all times which application, application window, or dialog has the input focus. If applications **1640** and **1642** each post a request to display an auxiliary screen image, it may be advantageous to have driver **1620** send the auxiliary image of application **1640** to touch screen **1610** only when application **1640** has the input focus. Similarly, driver **1620** sends the auxiliary image of application **1642** to the touch screen only when application **1642** has the input focus. If application **1644** has not posted an auxiliary image, then when application **1644** has the input focus, driver **1620** may displays a default iconic screen image similar to that of **FIG. 4**.

[0124] When the user touches the touch sensor, driver **1620** forwards the finger touch information to the application with the input focus if that application has posted an auxiliary screen image that overrides the default finger motion behavior. Similarly, driver **1620** forwards finger tapping information to the application with the input focus if the application has posted an auxiliary screen image that overrides the default finger tapping behavior.

[0125] Driver **1620** also monitors the keyboard, touch screen, or other devices to implement the various touch screen activation mechanisms disclosed in relation to **FIGS. 6A and 6B**. If the auxiliary screen of an application is displayed, the driver conveys the activation state to the application to allow the application to interpret finger motions and taps correctly. If the default iconic screen is displayed, the driver uses the activation state to decide whether to forward motion or tapping information about the icon under the cursor to the application that posted the icon.

[0126] Pop-up screens may be created using similar means to auxiliary screens. However, for pop-up screens, driver **1620** may advantageously maintain a concept of touch screen pop-up focus distinct from the input focus maintained by the operating system for applications on the main display. Driver **1620** must use some reasonable rule to coordinate between multiple requests for auxiliary and pop-up images.

[0127] Driver **1620** may implement some icons, auxiliary screens, and pop-up screens entirely within the driver itself. The driver may include a mechanism for associating auxiliary screens with pre-existing applications that do not recognize API **1624**. For example, if a known pre-existing presentation software application has the input focus, the driver could supply an auxiliary screen like that of **FIG. 8A**. Driver **1620** would interpret taps in the special icons of **FIG. 8A** by sending corresponding signals known to be recognized by the software application, such as simulated keystrokes or mouse clicks. Driver **1620** may also implement a mechanism to allow users to associate special icons or auxiliary screens with pre-existing applications. One such mechanism is a scripting language including commands to display images and icons and to change the interpretation of finger actions; scripts in such a language could be written and associated with a software application without modification to the application itself. Another such mechanism is a graphical control panel similar to the resource editors present in many interactive programming environments.

[0128] Driver **1620** may support a software control panel to allow the user to customize the operation of the touch screen. This control panel can include controls to choose, enable, disable, and rearrange the various icons on the default iconic screen. This control panel can also include controls to choose which touch screen activation mechanism(s) to use, and which auxiliary and pop-up images to allow access to the screen.

[0129] Driver **1620** may allow combinations of iconic, auxiliary and pop-up images on the touch screen. For example, driver **1620** could implement a concept of overlapping windows on the touch screen whereby an auxiliary screen could overlap part of but not all of the iconic screen image it replaces. One possible implementation of this approach is to use the existing display driver architecture of the operating system to manage the display of the touch screen. In the most general case, the touch screen would be viewed as a second display monitor by the operating system, and applications could open windows and dialogs on this display just as they would on the main display.

[0130] In an alternate embodiment, the touch screen would be treated distinctly from the main display. Applications would be forbidden from opening windows on the touch screen or operating the touch screen by means other than API **1624**. This approach is less flexible but more appropriate, as the small size of the touch screen causes it to behave in the computer/human interface as a different class of device than main displays, even though the touch screen and main display might both be implemented by similar raster LCD technology.

[0131] While the present invention has been described with reference exemplary embodiments, it will be understood by those skilled in the art that various changes may be made and equivalents may be substituted for elements thereof without departing from the scope of the present invention. In addition, many modifications may be made to adapt a particular situation or material to the teachings without departing from the essential scope thereof. Therefore, it is intended that the present invention not be limited to the particular embodiment disclosed as the best mode contemplated for carrying out this present invention, but that the present invention will include all embodiments falling within the scope of the appended claims.

What is claimed is:

1. In a computing device having a processor running an operating system and a display, a graphical user interface, comprising:

a touch screen; and

a driver coupling said touch screen to said operating system, said driver displaying a plurality of icons on said touch screen, at least one of said icons identifying at least one region on said touch screen that will cause an action on said display and not on said touch screen in response to contact by an object on said region.

2. The graphical user interface of claim 1, wherein at least one of said icons identifies at least one other region on said touch screen that will cause another action on said touch screen and not on said display in response to contact by said object on said other region.

3. The graphical user interface of claim 1, wherein at least one of said icons identifies at least one other region on said touch screen that will cause another action on both said display and said touch screen in response to contact by said object on said other region.

4. The graphical user interface of claim 1, wherein the processor drives a default screen image onto said touch screen.

5. The graphical user interface of claim 4, wherein said default screen image includes at least one cursor positioning region on said touch screen for moving a cursor on said display in response to said object touching said touch screen.

6. The graphical user interface of claim 4, wherein said default screen image includes at least one action control icon identifying a region on said touch screen for causing an action on at least one of said display and said touch screen in response to said object touching said touch screen.

7. The graphical user interface of claim 4, wherein said default screen image includes at least one of a vertical scroll bar icon and a horizontal scroll bar icon identifying regions

on said touch screen for scrolling displays in a window on said display screen in response to said object touching said touch screen.

8. The graphical user interface of claim 1, further comprising a secondary pointing device.

9. The graphical user interface of claim 8, wherein said secondary pointing device is chosen from the group comprising an isometric joystick and an external mouse.

10. In a computing device having a processor running an operating system and a display, a graphical user interface, comprising:

a touch screen, said touch screen supporting an unactivated state and an activated state; and

a driver coupling said touch screen to said operating system, said driver displaying plurality of icons on said touch screen, at least one of said icons identifying at least one region on said touch screen that will cause an action on said display and not on said touch screen in response to contact by an object on said region.

11. The graphical user interface of claim 10, wherein at least one of said icons identifies at least one other region on said touch screen that will cause another action on said touch screen and not on said display in response to contact by said object on said other region.

12. The graphical user interface of claim 10, wherein at least one of said icons identifies at least one other region on said touch screen that will cause another action on both said display and said touch screen in response to contact by said object on said other region.

13. The graphical user interface of claim 10, wherein the processor drives a default screen image onto said touch screen.

14. The graphical user interface of claim 13, wherein said default screen image includes at least one cursor positioning region on said touch screen for moving a cursor on said display in response to said object touching said touch screen.

15. The graphical user interface of claim 13, wherein said default screen image includes at least one action control icon identifying a region on said touch screen for causing an action on at least one of said display and said touch screen in response to said object touching said touch screen.

16. The graphical user interface of claim 13, wherein said default screen image includes at least one of a vertical scroll bar icon and a horizontal scroll bar icon identifying regions on said touch screen for scrolling displays in a window on said display screen in response to said object touching said touch screen.

17. The graphical user interface of claim 10, wherein a visual characteristic of at least one of said icons is modified when said touch screen is in said activated state.

18. The graphical user interface of claim 17, wherein said visual characteristic is a change in coloration.

19. The graphical user interface of claim 17, wherein said visual characteristic is an outline surrounding at least one of said icons.

20. The graphical user interface of claim 10, wherein said activated state and said unactivated state are controlled by pressing a key on a keyboard.

21. The graphical user interface of claim 20, wherein said key is on a standard keyboard layout.

22. The graphical user interface of claim 10, further comprising a secondary pointing device.

23. The graphical user interface of claim 22, wherein said secondary pointing device is chosen from the group comprising an isometric joystick and an external mouse.

24. The graphical user interface of claim 10, wherein at least one of said icons is touch-sensitive in said unactivated state.

25. The graphical user interface of claim 10, wherein at least one of said icons is located at an edge or a corner of said touch screen.

26. The graphical user interface of claim 10, wherein at least one of said icons is sensitive to a tapping motion when said touch screen is in said activated state.

27. The graphical user interface of claim 10, wherein at least one of said icons is sensitive to motion by said object when said touch screen is in said activated state.

28. The graphical user interface of claim 10, wherein said activated state and said unactivated state are controlled by pressing a button.

29. The graphical user interface of claim 10, wherein said activated state and said unactivated state are controlled by touching a touch sensitive part of said computing device.

30. The graphical user interface of claim 10, wherein said activated state and said unactivated state are controlled by touching at least one of said icons on said touch screen.

31. The graphical user interface of claim 10, wherein said activated state and said unactivated state are controlled by touching at least two objects at once to said touch screen.

32. The graphical user interface of claim 10, wherein said activated state and said unactivated state are controlled by tapping two or more times in rapid succession in said region on said touch screen.

33. The graphical user interface of claim 10, wherein said activated state and said unactivated state are controlled by hovering over said region on said touch screen without touching said region.

34. The graphical user interface of claim 10, wherein said activated state and said unactivated state are controlled by holding said object steady in said region on said touch screen for a specified duration.

35. In a computing device having a processor running an operating system and a display, a graphical user interface, comprising:

a touch screen; and

a driver coupling said touch screen to said operating system, said driver displaying a plurality of icons on said touch screen, at least one of said icons identifying at least one region on said touch screen that will cause an action on said display and not on said touch screen in response to contact by an object on said region;

wherein said driver includes an application programming interface that enables an application to display at least one image on said touch screen.

36. The graphical user interface of claim 35, wherein at least one of said icons identifies at least one other region on said touch screen that will cause another action on said touch screen and not on said display in response to contact by said object on said other region.

37. The graphical user interface of claim 35, wherein at least one of said icons identifies at least one other region on said touch screen that will cause another action on both said display and said touch screen in response to contact by said object on said other region.

38. The graphical user interface of claim 35, wherein the processor drives a default screen image onto said touch screen.

39. The graphical user interface of claim 38, wherein said default screen image includes at least one cursor positioning region on said touch screen for moving a cursor on said display in response to said object touching said touch screen.

40. The graphical user interface of claim 38, wherein said default screen image includes at least one action control icon identifying a region on said touch screen for causing an action on at least one of said display and said touch screen in response to said object touching said touch screen.

41. The graphical user interface of claim 38, wherein said default screen image includes at least one of a vertical scroll bar icon and a horizontal scroll bar icon identifying regions on said touch screen for scrolling displays in a window on said display screen in response to said object touching said touch screen.

42. The graphical user interface of claim 35, wherein said application programming interface accepts user input from said touch screen.

43. The graphical user interface of claim 35, wherein said image includes at least one application icon.

44. The graphical user interface of claim 43, wherein said at least one application icon launches a software application.

45. The graphical user interface of claim 35, wherein said application programming interface activates a backlight of said touch screen to alert a user.

46. The graphical user interface of claim 35, wherein said application programming interface audibly alerts a user to a presence of said image.

47. The graphical user interface of claim 35, wherein said image is a magnified portion of said display.

48. The graphical user interface of claim 35, wherein said image assists an input method for ideographic characters.

49. The graphical user interface of claim 48, wherein said input method uses handwriting entry on said touch screen.

50. The graphical user interface of claim 35, wherein said image is a trail indicating a path of motion of said object.

51. The graphical user interface of claim 35, wherein said image is used for user authentication.

52. The graphical user interface of claim 35, wherein a visual characteristic of at least one of said icons is modified when said touch screen is in an activated state.

53. The graphical user interface of claim 52, wherein said visual characteristic is a change in coloration.

54. The graphical user interface of claim 52, wherein said visual characteristic is an outline surrounding at least one of said icons.

55. The graphical user interface of claim 35, further comprising a secondary pointing device.

56. The graphical user interface of claim 55, wherein said secondary pointing device is chosen from the group comprising an isometric joystick and an external mouse.

57. The graphical user interface as in any one of claims **1**, **10**, and **35**, wherein said touch screen displays a manufacturer's logo when said driver is not activated.

58. The graphical user interface as in any one of claims **1**, **10**, and **35**, wherein at least one of said icons is visible on said touch screen in response to user input.

59. The graphical user interface as in any one of claims **1**, **10**, and **35**, wherein at least one of said icons is visible on said touch screen in response to activating a key on a keyboard.

60. The graphical user interface as in any one of claims **1**, **10**, and **35**, wherein at least one of said icons is visible on said touch screen in response to activating a touch region.

61. The graphical user interface as in any one of claims **1**, **10**, and **35**, wherein at least one of said icons displays a manufacturer's logo.

62. The graphical user interface as in any one of claims **1**, **10**, and **35**, wherein at least one of said icons displays a keyboard lock state.

63. The graphical user interface as in any one of claims **1**, **10**, and **35**, wherein at least one of said icons displays battery status.

64. The graphical user interface as in any one of claims **1**, **10**, and **35**, wherein at least one of said icons is enlarged in response to said object touching at least one of said icons.

65. The graphical user interface as in any one of claims **1**, **10**, and **35**, wherein at least one of said icons is displaced in response to said object touching at least one of said icons.

66. The graphical user interface as in any one of claims **1**, **10**, and **35**, wherein at least one of said icons is selected by said object touching proximate to at least one of said icons.

67. The graphical user interface as in any one of claims **1**, **10**, and **35**, wherein said object is chosen from the group comprising a finger, a pen, a pencil, and a pointer.

68. The graphical user interface as in any one of claims **1**, **10**, and **35**, wherein at least one of said icons displays disk drive activity.

69. The graphical user interface as in any one of claims **1**, **10**, and **35**, wherein at least one of said icons is situated so as to be visible when a cover of the computing device is closed.

70. In a computing device having a processor running an operating system and a display, a graphical user interface, comprising:

a touch screen displaying a default screen image, wherein said image has at least one of a first set of icons displayed when a driver is loaded in said operating system and at least one of a second set of icons displayed when said driver is not loaded in said operating system.

71. The graphical user interface of claim **70**, wherein at least one of said second set of icons is a manufacturer's logo.

* * * * *