
Self-Revealing Gestures: Teaching New Touch Interactions in Windows 8

Kay Hofmeester

Microsoft
One Microsoft Way
Redmond, WA 98052
kayh@microsoft.com

Jennifer Wolfe

Microsoft
One Microsoft Way
Redmond, WA 98052
jewolf@microsoft.com

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

CHI'12, May 5–10, 2012, Austin, Texas, USA.

Copyright 2012 ACM 978-1-4503-1016-1/12/05...\$10.00.

Abstract

The touch language we use to interact with computers and devices is still developing. How can we teach users of our systems new touch gestures without interfering with their user experience? A team of user experience designers and researchers went through an iterative process to design a teaching method for two new touch interactions. This case study describes the designs they created, their insight from user studies, and the final design that will be implemented in Windows 8.

Author Keywords

Touch; gestures; learning; self-revealing gesture; design; feed forward; feedback; natural user interface

ACM Classification Keywords

H.5.2 [Information interfaces and presentation]: User Interfaces - Graphical user interfaces (GUI).

Introduction

Touch user interfaces are all the rage. Computer hardware manufacturers are introducing touch phones and tablets, software developers are creating touch applications, and the media discuss touch computing developments almost every day. Some think that touch

is a better way to interact with computers than a mouse because it is more natural.

But what does natural mean in this context? Some think it means intuitive, so users will not have to learn the gestures required to interact with the system. Some think natural interaction refers to real world interactions, which results in a user interface that imitates the physical world. We argue that natural interactions don't exist in a computer system [1,2]. We also believe that copying the real world in a computer user interface limits the flexibility of that environment. The goal of a computer system is to enable more and different interactions, not to copy the ones we can already perform in the physical world.

Instead our focus for Windows 8 was on finding gestures that were optimized for the touch input method and the user interface we were developing.

Each input method is best at something and worst at something else, determined by its properties [3,4]. Mouse for instance is an indirect input method that translates movement of the mouse on a desk into movement of the cursor on the screen with a certain gain. That is one of the characteristics which make it a very precise input method. However, to work properly a mouse requires that a flat tracking surface is available. Some other examples of the results of these properties are that the keyboard is great at text input and shortcuts, but not very good at random access navigation and touch is great at directly interacting with content, but not at being precise.

Simply adding support for an input method does not automatically create a great user experience. It is

important to design the user interface in a systematic way that takes the input methods' strengths and weaknesses into consideration. The concept of input languages can be used to help us create such a framework [5].

A user interface is the medium for people and computers to communicate. We speak to computers using input methods like keyboard, mouse or touch. The computer replies using output methods like the visuals on the screen and audio. This communication requires a language, which is a set of conventions on how to convey commands and information. Every software system that interacts with people has a language. When the keyboard was the single input method used to interact with computers, that language was the command line interface. When the mouse was invented, the language of the graphical user interface was introduced. And currently the software industry is working on defining touch language conventions. Some have been established, like pinch to zoom, but there are still different approaches to a touch language in today's computer systems, for instance for commanding, selection, and navigation.

For Windows 8 we defined new gestures that worked well for the properties of touch and helped to optimize interaction with the system. One of those gestures is "slide to select". This gesture enables a key action in the system, selecting one or multiple objects that can subsequently be acted on. We optimized this gesture for the properties of finger movement, it is easy to contract the muscles of the index finger and this can be done in a precise way. User research confirmed that this gesture and the result it invoked worked well in the Windows 8 user interface.

But since this gesture does not follow commonly known user interface conventions or copies an interaction from the real world, it was not easily discoverable for users. We had to find a way to teach the gesture to users while minimizing the mental effort of the user learning to interact with the new system.

This paper describes the design process a team of user experience designers and researchers followed to develop a way to teach users of Windows 8 a new gesture for selection of objects. We did not want this to be a tutorial, but rather we wanted users to learn the gesture while they are using the system. The initial design was informed by research and design insights from Microsoft Surface.

Touch Language

One of the goals of Windows 8 was to create a user interface that provides a great touch experience. As part of that effort, we developed a touch language which defined interactions and user interface behaviors for touch input [5]. At the beginning of our process we established goals for touch interaction:

- Simple and consistent
- Limited gesture set
- Direct manipulation
- Reversible interactions
- No modes
- Limited use of timed gestures

We also defined how the user would articulate the value of the experience as:

- Fast and fluid
- In control

- System responds as expected
- Consistent, engaging, functional and efficient

In order to create a touch optimized system, we needed to develop new gestures. These gestures were optimized according to our goals and they were optimized for the strengths of the touch input method. As discussed in the introduction, we did not optimize the gestures for discoverability. We'd rather define the optimal gesture for the interaction and then help users to easily learn those gestures. This posed a challenge: How do we teach Windows 8 users the new gestures without impairing the user experience?

Teaching New Gestures

There are many ways to teach users of software new functionality: manuals, help systems, tutorials, contextual tooltips, and even advertising, to mention a few. When designing the experience for teaching new gestures we had several principles.

First, we wanted the learning to happen when the user needs it [6]. We did not want to disrupt the experience of discovery by inserting a learning experience at a fixed point in the timeline, for instance after set-up, or when starting a new feature or application. Second, we did not want the user to have to specifically seek out information to learn this new gesture, such as accessing a manual or help system. Thirdly, we wanted to teach the new gestures without introducing new visual user interface elements, because we did not want this system to get in the way after the user had acquired the new gesture. Fourth, we wanted the learning system to be consistent across the product and across a variety of scenarios. Our goal was to create a learning experience for users to discover the new



Figure 1. Visual element for closing an application in the Surface design.

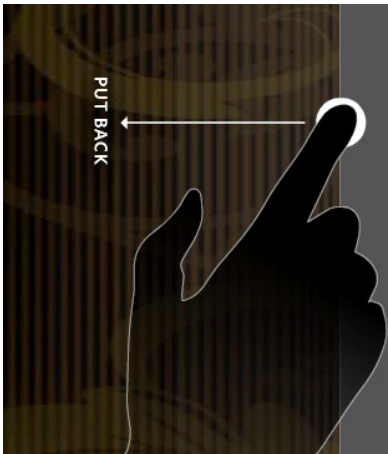


Figure 2. Finger down revealed gesture direction and distance threshold.

gestures while using the software, when the need for the functionality arises. This was similar to the goals of several other research and software development teams [7,8,9,10,11].

Surface Designs

Our initial design was based on the insights from the gesture teaching designs created for the Microsoft Surface version two user interface. Here follows a description of that design process and what we learned from it.

Users tend to tap on user interface elements, which is behavior they have learned with the mouse. This is evidenced by the observation that they often referred to tapping as “clicking” in the Surface user research. While clicking a mouse button works well for closing an application, tapping is not the optimal gesture for closing an application due to the ease of accidental activation on Surface’s vision-based system. If a user would for instance accidentally brush their sleeve over the closing button, that action would close the application. Instead the team focused on teaching users to use a more deliberate, but equally easy sliding gesture to close applications.

The team wanted to teach the new gesture to users while they were using the system and decided to leverage the natural tendency to tap in the gesture teaching design. When the user tapped on the visual element, which was a circle with an X, the user interface showed visual cues or feed forward to indicate the desired action to close the application. The visual cues consisted of an arrow to indicate the direction of the gesture and text on an angle perpendicular to the arrow to indicate the distance threshold the sliding

finger needed to cross to close the application. The design was evaluated in a user study with eight participants. The participants were asked to navigate the user interface. The tasks required them to open and close applications. They were not told how to perform this task, so the team could test if they could learn the interactions without prompting from the researcher. We learned the following:

- All participants tapped on the visual element and subsequently noticed the visual cues.
- Most participants learned the new closing gesture within several minutes without prompting. Two users needed prompting to continue to discover the new gesture.
- The visuals and timing of the animations needed small adjustments to improve the learning process.
- All participants were able to recall and reproduce the gesture to close other applications.
- When asked, participants preferred tapping to sliding to close the application.

In addition to improving the time needed to learn the new gesture, the team wanted participants to evaluate the gesture positively. By looking closer at the study results, the conclusion was that the preference for tapping was related to the circle with an X visual as the symbol for closing the application. This is very similar to the visual language used by Windows for closing a Window with the mouse pointer. The team concluded that users would compare clicking with a mouse to sliding the finger. Because they were used to clicking with a mouse, they indicated that was easier to perform. To test this conclusion, a visual element that looked distinctly different from the Windows visual was created. When this was tested, the conclusion was

found to be correct. The participants learned the new closing gesture within a minute and they expressed to like the interaction. After using the system for several minutes, they seemed to use the gesture without thinking about it, enabling them to focus on the applications.

We learned several things from this design effort. We learned it was possible to use a gesture to teach another gesture. We learned what kind of visual elements we could use for teaching and that it is important to create a visual language to match the new gestures, in order to avoid users falling back in old habits they learned in other software systems. Finally we learned the importance of carefully fine-tuning the detailed behavior of the learning method.

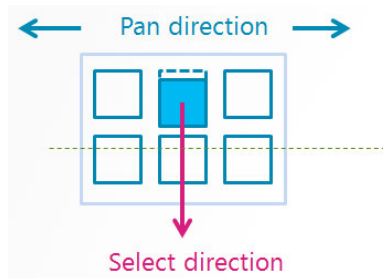


Figure 3. A diagram of the swipe to select gesture.

Swipe to Select

For Windows 8 we wanted to enable users to perform multiple actions on the content in the user interface, while minimizing the need for visual user interface elements that are continuously present and while minimizing mode switching. This required extending the common gesture set frequently used on most consumer touch devices. Similar to the Surface situation, we also wanted to minimize accidental activation of interactions.

For example, in an earlier study we had compared two different checkout paths for purchasing applications. One path allowed users to drag application tiles to a checkout basket. The second checkout path allowed users to click a button to purchase. All users preferred the direct action of dragging a tile to the basket despite the fact that it took more time than tapping a button to purchase. They enjoyed the direct and connected action

of drag to purchase and described it as “playful” and “fun”.

The rest of this paper focuses on the gesture for selection. Selection is an important interaction in Windows 8 that supports other interactions, because it enables manipulation of one or multiple objects in the system. We wanted this gesture to create an efficient interaction that would enable multi-select and still provide the feeling of a connected experience. From literature, Surface research and our own research we knew that direct manipulation was positively evaluated by users [12].

With this in mind we designed the “swipe to select” gesture. Swiping the finger orthogonally to the scrolling direction to select is quick, leverages the strengths of finger movement properties, and can be implemented as direct manipulation. When it is implemented as direct manipulation the interaction is reversible: the user can return the finger to the starting position and lift without having to complete the action. The result is an efficient gesture that would be a primary interaction across the system.

However, it would be necessary for users to learn that interaction within the first few sessions using the system. We set out to create a teaching mechanism that would guide the user to quickly discover the gesture.

Research Design

The process of creating the teaching method for the swipe to select gesture consisted of a series of iterative design and user research phases. Over the course of this process research the team developed several



Figure 4. A study participant with a tablet computer in the usability lab.

interactive prototypes to help us evaluate several aspects of the design:

- Do the specific mechanics of the interface work for users?
- Does the gesture get in the way or slow down the experience?
- How quickly do users discover and learn the gesture?
- How long does it take users to feel confident?
- Can users transfer their knowledge and apply the gesture across the system?
- What words do users use to describe the gesture and the touch language?

We conducted nine user research studies over the course of one year. The typical study design included five to eight adult participants aged 22 to 64 with varying degrees of touch interaction experience, from no touch experience to daily touch device use. Each individual session lasted approximately two hours.

While all studies were conducted in a standard usability lab, the furnishings in the lab were updated to reflect a comfortable living or sitting room. Furnishing included sofa, coffee table, side table, pillows and a blanket. All participants worked on touch tablet or touch laptop convertible and were told to relax, get comfortable and imagine they were at home.

At the beginning of each study participants were given time to interact with the device and get familiar with their content and the feeling of holding that particular device. This enabled us to see how confident and familiar they were with a touch device and what gestures were already in their vocabulary. After a few

minutes of warm-up the participants were asked to complete several tasks that required selecting items. For example: "You would like create a photo album of eight of your favorite photos and share the album with your best friend." Participants were given no hints or instruction as to how to complete the task. Participants were only given hints if significant time (five to seven minutes) had passed before they discovered the selection gesture or they had given up completing the task.

During each study we used the RITE methodology to iterate on the design for the self-revealing gesture for selection [13]. This enabled us to apply changes to our design based on research insights during the study and was especially helpful when fine-tuning the gesture.

Looseness

Our first design for creating a mechanism to teach the selection gesture was to show "looseness" during scrolling. We created a prototype of a horizontally scrolling page of images of music albums that could be selected to add to a queue. The intent was to demonstrate the affordance for the desired gesture in an easily discoverable manner. In most systems, when scrolling a page with content, the content only moves in the direction of the scrolling action. For instance, in a horizontally scrolling list the items in the list don't move up or down when a user moves their finger sideways on the list. Our design was to let the items move up and down to indicate the possibility of a swipe orthogonal to the scrolling direction. The natural motion of our hands, wrists and arms is an arc shape, not a straight line. Due to this, the looseness of the tile could be discovered when scrolling the page, because their anatomy makes users drag slightly down or upward in



Figure 5. Explorations for self-revealing gesture visuals.

an arc on the tile while scrolling horizontally. This demonstrated that an object the finger was on could also be moved in the orthogonal direction.

We tested this design in a user study. Users would attempt to select using methods learned from mouse or other touch devices: tap, double tap, press and hold, or looking for a graphical user interface widget. This behavior was observed in all of our user studies to varying degrees. After working through a series of known actions, the swipe to select gesture was discovered by over half of the users and repeated multiple times to select several objects.

However, some users did not understand why the album tiles were moving or what they were supposed to do: "I'm not sure what that is or what it means." The movement did not communicate that sliding an object up or down would result in selection. In addition, the animation provided a great degree of visual noise, was distracting to users, interfered with scrolling and did not visually work well on vertically panning surfaces. The animation we had included in the prototype did not reach the goals we had established for the teaching mechanism.

Despite the challenges learning the selection gesture, once users had learned the gesture all participants felt that dragging down to select was "fast and easy". Several commented that it was "easier than double-tap".

As a result of this study we understood that teaching animations and feed forward must hint at the intended outcome of the action. We also learned that although we wanted the teaching animation to be easy to

discover it should not interfere with or distract from the primary interaction. We did not want the secondary action of selection to negatively influence the primary action of scrolling. Because the looseness design and teaching mechanism was confusing to users, did not effectively teach the swipe to select gesture, and detracted from the user experience, we abandoned this design direction.

Hold to Learn

When thinking about teaching new gestures, we wanted to develop a consistent teaching mechanism or teaching gesture that users could apply across the system to learn and safely explore. Effectively this is what was done at Surface. In that design a learning gesture was created based on a common action for both touch and mouse users. By tapping on a visual element users would invoke visual elements that taught them how to perform the gesture needed to close an application. In the Windows system tap was already used for the primary action of launching or activating an object. We did not want to use double tap because of the resulting latency on the system; every time a user taps, the system has to wait for the second tap time threshold to pass before responding to the first tap.

We needed a simple way of invoking the teaching visuals or animations, or a "self-revealing gesture" [1,11]. Through our research we discovered that when using a new device, customers with prior touch experience will experiment or see what the device can do by applying their vocabulary of known gestures. Customers without touch experience will in turn use gestures that simulate mouse interactions.

For our next design direction we decided to use the “press and hold” gesture for teaching. We did not want to use press and hold for system actions because the timing of the gesture can never be optimized for the full user base. In some software systems, press and hold is used to invoke functions such as a context menu. However, if a user accidentally places their finger on the screen for too long, they will unintentionally invoke the context menu. For experienced users, the timed gesture results in having to wait for the menu every time they perform the gesture, slowing down the user.

For our purpose, accidental invocation of learning visuals was not a problem, because it would not stop the user from performing other actions. Once the selection gesture was learned the user would no longer need to apply the learning gesture. Also, since the process of learning primary actions usually occurs during in the initial stages of the software engagement, slowing the user down did not pose as much of a problem. Because of these reasons we decided to pursue the design of “press and hold to learn” behavior.

Self-revealing gesture

Our objective was to leverage the known press and hold gesture, frequently used in other devices and similar to mouse hover, to teach the new swipe to select gesture. The next step was to design the on-screen behavior that results when a user presses and holds their finger on an object. We tried out a range of different visuals and animations including arrows and lines indicating thresholds with the goal of teaching:

- Slide the object in the indicated direction
- Slide the object across the indicated distance threshold to select

Early in the design process we discovered that these two factors had a major impact on what and how we could teach the swipe to select gesture using on-screen visuals. Visuals such as arrows or threshold lines that appeared below the object the user is interacting with or at the right bottom corner of the screen were either easily overlooked or occluded by the fingers and hands.

The team created a new interactive prototype of a photo application. When a user pressed and held on a photo it would shift down slightly, revealing an arrow pointing down. The downward motion of the tile and arrow were intended to communicate “drag the object down to select”.

We tested this prototype in a user study. As before, our goal was to quickly respond to user issues by iterating on the design between sessions using the RITE method. After performing a few navigation tasks users were asked to select several photos to add to a slideshow. The first two participants discovered the arrow at press and hold but did not understand what it represented. Because of this they struggled for a few minutes to discover how to select an item. One participant grew increasingly frustrated during this process and kept searching for a GUI control, tapping or double tapping without success. Several times she commented she “did not know what the arrow means” and that she was “stuck”. The other participant was able to discover how to select through a process of “trial and error”. Our design was not as intuitive or as discoverable as desired for a teaching gesture.

Because the first two users noticed the arrow at press and hold but did not understand what it represented, we added the text “select” to the left of the arrow. But

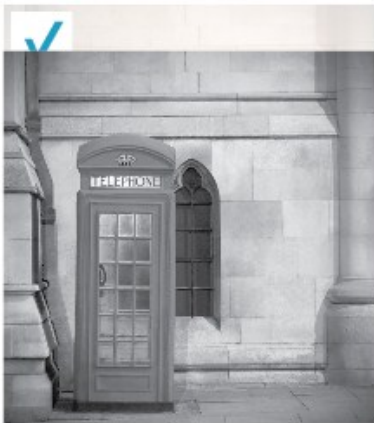


Figure 6. The three object states: Unselected, during selection gesture, and selected

the research showed that many users did not read the text or did not understand the relationship between the word “select” and the arrow pointing down.

Additionally we discovered that users without touch experience are not familiar with the gesture press and hold. Some would attempt the gesture on their own through a process of exploration. Other non-touch users did not have this gesture in their vocabulary and therefore needed a hint to apply the gesture and discover the selection interaction. Despite the challenges learning the selection gesture all but two participants (one with prior touch experience, one without) were able to eventually select and unselect items without being prompted by the user researcher. User description of the gesture ranged from “different” and “not as easy as expected” to “gets easier after a few times” and “fun”.

Throughout this study and subsequent studies described below, most of the participants went through four phases of discovery and learning:

- 1) uncertainty
- 2) discovery of the intended behavior and skepticism
- 3) understanding and skill development
- 4) confidence and acceptance or enjoyment.

Moving the object down slightly at press and hold worked to get the users’ attention and demonstrated that the object could move in the indicated direction. We had designed a visual affordance of swiping down, now we needed to communicate the result of swipe: selection of the object and appearance of contextual commands. After several design iterations we chose to place a checkmark in the upper right corner of the

object. This icon was based on Windows conventions, since it represents selection in Windows 7.

We updated the photos prototype to include the checkmark icon. During the first study with the checkmark design we again asked users to add photos to a photo album. Several participants recognized the icon, understood what it represented and were able to select an item. One participant recognized the checkmark and thought that the item had been selected at press and hold. They did not realize their error until after they had attempted to select several items but did not have any photos in their basket. They were able to successfully select photos after several more attempts. “I drag it down until I get the little check in it”. Other participants did not understand what the checkmark represented. Despite this all but one participant was able to successfully select an item without direction from the user researcher.

In the second study with the checkmark icon we added the text “select” next to the checkmark. Once again, most participants did not notice, read the text or recognize the relationship between the word “select” and the checkmark. The combination of the checkmark and the word “select” did not communicate that they needed to select the item to invoke a command bar menu containing a list of contextual actions (such as share or slideshow) necessary to complete the required task. Also, because the command bar menu did not appear until after the item was selected, users did not learn that the checkmark and command bar were related. They only discovered this relationship once an item had been successfully selected. “Oh, okay I selected by moving down. That makes sense.”



Figure 7. The checkmark visual during and after the interaction.

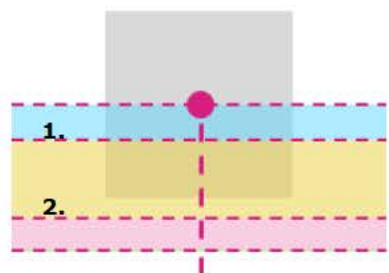


Figure 8. The thresholds for the selection gesture: 1. Selection, 2. Resistance.

Next we updated the photos prototype by combining the shifting down animation with a checkbox that was hidden behind the object, removed the word “select”, and tied the selection gesture to the invocation of the command bar. When the user pressed and held, the object slid down partially revealing the checkbox. When the users pulled the object down further towards the selection threshold, more of the checkbox was revealed and the entire command bar came into view during the interaction.

After several attempts at selecting an item using tap or double tap, participants would press and hold on the object and notice that something was hidden behind the object. Most participants would follow the animation with their finger by sliding the object down. At this point they would also see the command bar appear as the checkbox was revealed. The combination of a recognizable symbol and the appearance of the command bar communicated the action needed to select an item. Within one to five minutes all users were able to select items on their own. Users who took longer to successfully select, commented that it was difficult for them to understand how far they needed to drag the item down.

After users were able to select several photos and add them to a slideshow, we asked them to unselect a few photos. Several users first attempted to tap on the checkmark to unselect. When that did not work they were able to discover that they could swipe to unselect as well. Within minutes of learning the correct select and unselect gesture users felt it was easy and demonstrated the ability to quickly select and unselect items. Even when users struggled to learn and become confident in the interaction, by the end of the session

they had forgotten this. Several users commented that selecting it was “easier than double tap”.

Fine-tuning of the gesture

After deciding the design direction for the learning gesture, the animations and the visuals, we needed to fine-tune the gesture itself. We divided the gesture in several stages:

- The animation invoked by finger down
- The object following the finger
- The object showing resistance to being dragged further
- The object returning to the original state after release of the finger

We learned that slight modifications to the timing and distance of each stage made a significant impact on the discoverability of the gesture and how the gesture felt. If the animation at press and hold was too short and did not reveal enough of the checkmark, the motion was easily overlooked. If it was too long, participants thought that simply pressing down selected the item.

Tuning was especially important in the resistance phase of the gesture and animation. This was designed to limit the “looseness” described earlier allowing for a better sense of control. Too much resistance made selection gesture feel “slow” or like “a lot of work”, especially when selecting multiple items in succession. When we applied the right combination of timing, distance and resistance study participants described selecting as a “quick flick”, “fast” “easy” and “fun”.

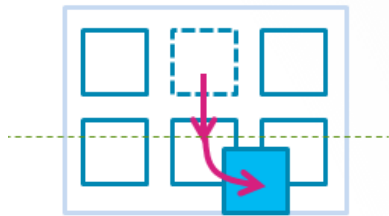


Figure 9. A diagram of the swipe to move gesture.

Selection and rearrange

Early in the development process for Windows 8, it became clear we needed a simple and direct way to move or rearrange objects within a collection. Because we wanted this functionality to be accessible without visual user interface elements and without invoking a mode, we decided to extend the “swipe to select” gesture. This required us to update the fine-tuning of the swipe gesture for this secondary purpose. In the new design, a short swipe would select the object, continuing the sliding gesture past a distance threshold would break the object loose from its grid so that it could be dragged to another location.

The results of testing this new combination of interactions showed the introduction of the move function impacted the discoverability of selection. Users with touch experience were familiar with the concept of drag to move and not familiar with our new gesture swipe to select. Because the downward shift of the object and partial reveal of the checkbox only appeared on press and hold, the self-revealing gesture for swipe to select was easily overlooked during drag to move. Study participants would slide the object down, past the selection threshold without noticing the selection checkbox that briefly appeared. They discovered the move interaction, but not the select interaction. Only through additional prompting by the researcher would users discover the selection gesture. From several users we heard comments like “I would not have found that on my own” and that it was “awkward”. The addition of rearrange had significantly limited the users’ ability to learn the selection gesture. We went back to the drawing board to create alternative versions for the self-revealing gesture.

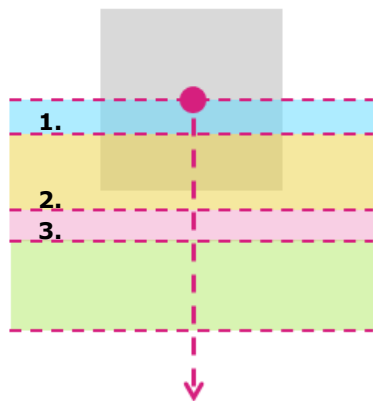


Figure 10. The thresholds for the combined gestures: 1. Selection, 2. Resistance, 3. Move.

We investigated using stronger visual cues such as arrows, text, and drop targets. None of these worked well enough to teach users the selection gesture consistently in user studies. Most of these designs were more confusing than the original design. In addition to overlooking selection, these designs caused users to have difficulty moving objects. Participants who were familiar with touch had learned from other devices to press and hold to enter a mode and then drag the tile to move it in any direction. Our approach allowed users to skip the waiting time of press and hold but required them to drag straight down, past the selection threshold over the rearrange threshold. This concept was difficult for users to learn on their own. They would press and hold, see the tile shift and assume they were in rearrange mode and then attempt to drag in any direction. Sometimes they were successful and other times they could not move the tile, but would scroll instead. Only after several minutes of trying to rearrange objects in the prototype were they able to move objects as desired.

Because none of our new designs were as effective as earlier prototypes, we decided to thoroughly investigate the original design and determine which variables had the most impact. Our goal was to fine-tune the self-revealing gesture and the gesture variables for selection and move so that the visuals and interactions would not compromise the discoverability of either interaction.

First we made it easier to rearrange by widening the angle at which the object could be dragged to rearrange. This allowed participants to more easily rearrange, but they found it harder to select. When trying to select participants would accidentally move

the object. It took practice to "get the hang of it" and "seems very particular to get it where it is selected". Although participants felt like they could not consistently select they enjoyed the action of quickly selecting or moving the objects on screen. Participants referred to the action as "Cool but seems like I have to be very careful to get it right in order to get the checkmark. My margin of error is very tiny" and "Really cool if I didn't have to be quite so precise".

In response to these comments we increased the resistance feedback to make it easier to notice when the object was selected. After passing the resistance threshold, the object would move slower than the finger. The intent of slowing the movement of the object while in the resistance zone was to separate the two actions. Participants noticed the object would "slow down" just after the contextual commands began to appear on screen. They realized when the object slowed down it signaled that they had selected the object.

This change made it slightly harder to get into the move zone but we had compensated for this by widening the angle for the move gesture. Once the user entered the move zone the object moved back under the finger to indicate the object had broken away from the grid. Finally we changed the timing for activation of the self-revealing gesture, so the object would move down sooner after pressing and holding on the object.

The effect of these changes was that both gestures and the self-revealing gesture worked much better. Changes made to the thresholds and visual feedback during this study improved the discoverability and ease of moving objects. Users still needed to drag the object

orthogonal to the scrolling direction first, but the action was less restricted and the timing of the visual feedback provided improved awareness of the resulting action and when the move threshold was reached. It was easier for participants to discover the dual functionality.

Unfortunately the design changes made to improve the discoverability of the gesture came with a cost. Some study participants thought they had to press and hold before they could slide to select. Some thought they had to select an item before it could be rearranged. While teaching users how to select, we inadvertently taught them an extra, unnecessary gesture. Our goal of providing a fast and fluid non-modal experience had not been reached.

These findings drove the final changes to the teaching method. Because the study participants thought they had to use press and hold to enter a selection mode, we needed to make sure the animation resulting from the press and hold action did not look like a persistent change. Instead of the object sliding down when the press and hold time threshold was reached, we changed the animation so the object would slide down and back up while the finger was still on the object. Our goal was for the teaching animation to demonstrate that the object could move but that the movement was not a fixed state or mode change. The animation also revealed the checkbox icon behind the object.

The team updated the design and created a new end-to-end prototype representing several applications in one system. We returned to the lab and asked participants to complete similar selection and rearranging tasks as before. Because the object would

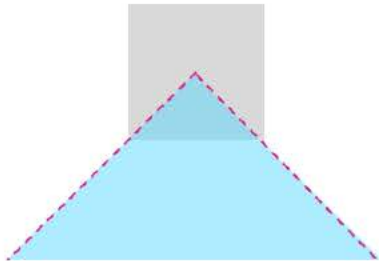


Figure 11. Drag angle for moving an object.

slide back up, most understood that press and hold did not invoke a mode. The feed forward animation was described as “it told me what I could do”. As a result of these changes, almost all participants were able to discover selection and rearrange in a manner that was not dependent upon a mode or persistent visuals and was fast and fluid. Some users with touch experience continued to exhibit the press and hold modal behavior at the beginning of the study, but by the end of the study, through a process of quickly selecting and unselecting multiple items, they were able to learn that press and hold was not necessary. Although this assumption does not impact the users’ ability to select and rearrange, it does impact the ease and efficiency of performing the gesture. Once they realized that press and hold was not necessary, several commented that it was “faster and more efficient” than their other device. They liked that they did not have to press-and-hold and wait to act on an item.

After successfully selecting and rearranging objects in one application we asked users to complete similar tasks in other applications included in the prototype. All participants were able to either apply the learning gesture press and hold to discover the self-revealing gesture and remind them of swipe to select or they were able to automatically apply the gesture without

the learning gesture. By the end of the study participants described the selection gesture as a “quick flick”, and described both selection and rearrange as “easy”, “effortless”, and “fast”. Users commented that they liked the consistency of the touch language and the ability to learn a gesture in one app and then apply it in other places across the system.

With this final update to the design of the self-revealing gesture, we created a teaching method that teaches users new to the system how to select an object using a gesture that was previously unknown to them.

Next steps

The self-revealing gesture discussed in this case study will be included in Windows 8. Once Windows 8 reaches the marketplace we will be able to determine how effectively this teaching method will work under various circumstances and with kinds of users not evaluated in the lab or encountered in field studies. The knowledge we gain can guide us to refine the design for this and future self-revealing gestures. It will also help us gain a better understanding of these teaching methods which can help us when we add teaching methods for other gestures. In addition to this, we propose a wider investigation into other invocation methods for self-revealing gestures.

Acknowledgements

We thank Adam Barlow, Moneta Ho-Kushner, Paul Hoover, Jan-Kristian Markiewicz, Jennifer McCormick, Leslie McNeil, Chantal Leonard, Trish Miner, Orry Soegiono, Chris Whytock, Daniel Wigdor, and Dennis Wixon.

References

- [1] Wigdor, D. and Wixon, D. *Brave NUI World: Designing Natural User Interfaces for Touch and Gesture*. Morgan Kaufmann (2011), 208, 145-156.
- [2] Wobbrock, J., Morris, M., and Wilson, A. User-defined gestures for surface computing. In *Proceedings of the 27th international Conference on Human Factors in Computing Systems*. CHI '09. ACM, New York, NY, 1083-1092. DOI=<http://doi.acm.org/10.1145/1518701.1518866>
- [3] Buxton, B. *Multi-touch systems I have known and loved*. <http://www.billbuxton.com/multitouchOverview.html>, 2011.
- [4] Buxton, B. *Sketching User Experiences: Getting the Design Right and the Right Design*. Morgan Kaufmann (2007).
- [5] Hofmeester, K. and Markiewicz, J.K. *Designing Metro Style Apps that are Touch Optimized*. <http://channel9.msdn.com/Events/BUILD/BUILD2011/APP-391T>
- [6] Catrambone, R. and Carroll, J. Learning a Word Processing System with Training Wheels and Guided Exploration. In *Proceedings SIGCHI/GI conference on Human factors in computing systems and graphics interface*. ACM (1987).
- [7] Bragdon, A., Uguray, A., Wigdor, D., Anagnostopoulos, S., Zeleznik, R., and Feman, R. Gesture Play: Motivating Online Gesture Learning with Fun, Positive Reinforcement and Physical Metaphors. In *Proceedings ITS 2010: Meta Gestures*, ACM Press (2010) 39-48.
- [8] Brandl, P., Forlines, C., Wigdor, D., Haller, M., and Shen, C. Combining and measuring the benefits of bimanual pen and direct-touch interaction on horizontal interfaces. In *Proceedings AVI 2008*, ACM Press (2008), 154-161.
- [9] Hinckley, K., Pausch, R., Proffitt, D., and Kassell, N.F. Two-handed virtual manipulation. In *Transaction on Computer-Human Interaction Volume 5, Issue 3*. ACM Press (1998), 260-302.
- [10] Hinckley, K., Yatani, K., Pahud, M., Coddington, N., Rodenhouse, J., Wilson, A., Benko, H., and Buxton, B. Pen + touch = new tools. In *Proceedings of the 23rd annual ACM symposium on User interface software and technology (UIST '10)*. ACM Press (2010), 27-36.
- [11] Freeman, D., Benko, H., Morris, M. and Wigdor, D. ShadowGuides: Visualizations for In-Situ Learning of Multi-Touch and Whole-Hand Gestures. In *Proceedings Conference on Interactive Tabletops and Surfaces 2009*. ACM (2009), 165-172.
- [12] Schneiderman, B. *Designing the User Interface strategies for Effective Human-Computer Interaction*. Addison-Wesley (1987).
- [13] Medlock, M., Wixon, D., McGee, M., & Welsh, D. The Rapid Iterative Test and Evaluation Method: Better Products in Less Time. In Bias, G., & Mayhew, D. (Eds.), *Cost Justifying Usability* Morgan Kaufmann (2005), 489-517.