

SUPPORTING TRANSITIONS TO EXPERTISE IN HIDDEN TOOLBARS

A Thesis Submitted to the
College of Graduate and Postdoctoral Studies
In Partial Fulfillment of the Requirements
For the Degree of Master of Science
In the Department of Computer Science
University of Saskatchewan
Saskatoon, Canada

By

KATHERINE SCHRAMM

© Copyright Katherine Schramm, January, 2019. All Rights Reserved.

PERMISSION TO USE

In presenting this thesis in partial fulfilment of the requirements for a Postgraduate degree from the University of Saskatchewan, I agree that the Libraries of this University may make it freely available for inspection. I further agree that permission for copying of this thesis in any manner, in whole or in part, for scholarly purposes may be granted by the professor or professors who supervised my thesis work or, in their absence, by the Head of the Department or the Dean of the College in which my thesis work was done. It is understood that any copying or publication or use of this thesis or parts thereof for financial gain shall not be allowed without my written permission. It is also understood that due recognition shall be given to me and to the University of Saskatchewan in any scholarly use which may be made of any material in my thesis.

Requests for permission to copy or to make other use of the material in this thesis in whole or part should be addressed to:

Head of the Department of Computer Science
176 Thorvaldson Building, 110 Science Place
University of Saskatchewan
Saskatoon, Saskatchewan S7N 5C9 Canada

OR

Dean
College of Graduate and Postdoctoral Studies
University of Saskatchewan
116 Thorvaldson Building, 110 Science Place
Saskatoon, Saskatchewan S7N 5C9 Canada

ABSTRACT

Hidden toolbars are becoming common on mobile devices. These techniques maximize the space available for application content by keeping tools off-screen until needed. However, current designs require several actions to make a selection, and they do not provide shortcuts for users who have become familiar with the toolbar. To better understand the performance capabilities and tradeoffs involved in hidden toolbars, we outline a design space that captures the key elements of these controls and report on an empirical evaluation of four designs. Two of our designs provide shortcuts that are based on the user's spatial memory of item locations. The study found that toolbars with spatial-memory shortcuts had significantly better performance (700ms faster) than standard designs currently in use. Participants quickly learned the shortcut selection method (although switching to a memory-based method led to higher error rates than the visually-guided techniques). Participants strongly preferred one of the shortcut methods that allowed selections by swiping across the screen bezel at the location of the desired item. This work shows that shortcut techniques are feasible and desirable on touch devices and shows that spatial memory can provide a foundation for designing shortcuts.

ACKNOWLEDGMENTS

I would like to express my gratitude to my supervisor Carl Gutwin for his guidance and input throughout my academic career at the University of Saskatchewan. I would also like to thank Andy Cockburn and Ashley Coveney for their assistance in this project. A big thank you to the faculty and staff at the Department of Computer Science, particularly the students and staff of the Human-Computer Interaction Lab for their help and friendship. Finally, I would like to thank my parents and sister; I could not have completed this project without their enthusiastic and unwavering support.

CONTENTS

<u>PERMISSION TO USE</u>	<u>I</u>
<u>ABSTRACT</u>	<u>II</u>
<u>ACKNOWLEDGMENTS</u>	<u>III</u>
<u>CONTENTS</u>	<u>IV</u>
<u>LIST OF TABLES</u>	<u>VII</u>
<u>LIST OF FIGURES</u>	<u>VIII</u>
<u>1 INTRODUCTION..... 1</u>	<u>1</u>
1.1 Introduction.....	1
1.2 Research Problem	3
1.3 Solution.....	3
1.4 Steps in the Solution	3
1.4.1 Create a framework for hidden toolbar designs.....	3
1.4.2 Implement a menu system using hidden toolbars	4
1.4.3 Test and evaluate hidden toolbars interactions in a menu system	4
1.5 Evaluation	4
1.6 Contributions.....	5
1.7 Thesis Outline	5
<u>2 RELATED WORK..... 7</u>	<u>7</u>
2.1 Hidden Toolbars.....	7
2.2 Command Selection	8
2.2.1 Toolbars	9
2.2.2 Crossing	11
2.2.3 Bezels.....	12
2.3 Transitions to Expertise	15
2.3.1 Motor domain.....	16
2.3.2 Spatial memory	17
2.3.3 Memorization.....	18
<u>3 EARLY DESIGNS AND PILOT STUDIES..... 20</u>	<u>20</u>

3.1	Initial Designs	20
3.1.1	Click.....	21
3.1.2	Swipe.....	22
3.2	Informal Evaluation	23
3.3	Second Designs.....	24
3.3.1	ButtonOpen.....	24
3.3.2	Timeout.....	25
3.3.3	DoubleTap.....	25
3.3.4	Swipe.....	26
3.4	Pilot Study of Second Designs.....	27
3.4.1	Methods.....	27
3.4.2	Results.....	28
3.5	Interpretation and Observations.....	30
4	<u>A DESIGN FRAMEWORK FOR HIDDEN TOOLBARS</u>	35
4.1	Design Goals.....	35
4.2	Hidden Toolbar Design Considerations.....	36
4.2.1	Mechanisms for indicating toolbar availability	36
4.2.2	Mechanisms for exposing the toolbar	36
4.2.3	Mechanisms for selecting (or cancelling) item selection.....	37
4.2.4	Shortcut selection.....	37
4.2.5	Learnability of items	38
4.3	Final Hidden Toolbar Designs	38
4.3.1	ButtonOpen.....	38
4.3.2	Timeout.....	40
4.3.3	DoubleTap.....	41
4.3.4	Swipe.....	42
5	<u>STUDY: COMPARING 4 HIDDEN-TOOLBAR DESIGNS.....</u>	44
5.1	Methods.....	44
5.1.1	Experimental Conditions	44
5.1.2	Procedure	45
5.1.3	Participants and Apparatus	45
5.1.4	Design and Research Questions.....	46
5.2	Results.....	46
5.2.1	Data Analysis	46
5.2.2	Overall Selection Time	47
5.2.3	Incorrect Selection Rate.....	48
5.2.4	Learning Item Locations.....	49
5.2.5	Transition to Accelerated Method	49
5.2.6	Effect of Item Locations on Learning and Performance.....	50
5.2.6.1	Effects on Selection Time	51
5.2.6.2	Effects on Learning	51
5.2.7	Subjective Results and Comments.....	52

6	DISCUSSION.....	56
6.1	Analysis of Research Questions.....	57
6.1.1	Q1. Are shortcut selection techniques faster?.....	57
6.1.2	Q2. Do people learn the item locations, and do they make a transition to the shortcuts?	57
6.1.3	Q3. Are error rates higher with memory-based shortcuts?.....	58
6.1.4	Q4. How does location affect learning and performance?.....	58
6.1.5	Q5. Which technique will be subjectively preferred?.....	59
6.2	Generalizing and Future Work.....	59
6.3	Conclusions.....	62
REFERENCES.....		63
APPENDIX A INITIAL DESIGNS STUDY.....		67
A.1	Consent Form.....	67
A.2	Demographic Form.....	69
A.3	TLX Forms.....	70
APPENDIX B SECOND DESIGNS STUDY.....		73
B.1	Consent Form.....	73
B.2	Demographic Form.....	75
B.3	TLX Forms.....	76
APPENDIX C FINAL DESIGNS STUDY.....		79
C.1	Consent Form.....	79
C.2	Demographic Form.....	81
C.3	TLX Forms.....	84

LIST OF TABLES

Table 3.1: All trials.....	29
Table 3.2: Trials after removing outliers.....	30
Table 5.1: Differences in user actions for the four techniques.....	44
Table 5.2: Mean (s.d.) effort questionnaire scores (1-5 scale, low to high), Friedman χ^2 value, and p-value.....	53
Table 5.3: Counts of participant preferences	54

LIST OF FIGURES

Figure 1.1:	Swiping across the bezel edge activates Android’s navigation drawer	1
Figure 1.2:	Swiping across the bezel edge activates Windows 8’s “charm”	2
Figure 2.1:	Display-edge hidden toolbars in current software systems. Swiping across the bezel edge activates Android’s navigation drawer (top left) and Windows 8’s “charms” (top right). Clicking on a button exposes Visual Studio’s “toolbox” (bottom).....	8
Figure 2.2:	A Hotbox widget, as it would appear in the centre of a screen, using a linear drop-down menu	10
Figure 2.3:	Hinckley <i>et al.</i> placed a narrow bar at the bezel at the top of the screen (a), when the user swipes down from the menu it expands the bar to a full menu (b). The user can then tap menu items (c) to create items and collapse the menu (d).....	11
Figure 2.4:	The CrossY drawing tool. A single motion selects both the colour and width for a pen stroke with no deceleration required in the targeting	12
Figure 2.5:	Three examples of Bezel Swipe gestures. (Left) A swipe in from the bezel at A to B on the screen selects an image. (Centre) A swipe in from the bezel A to B on the screen starts a text selection. (Right) A swipe in from C on the bezel to D on the screen ends the text selection.....	13
Figure 2.6:	(Left) A tapping gesture centred on the target at the edge of the screen may not successfully activate it but a swipe starting with a touch centred on the bezel (Right) is more likely to do so.....	14
Figure 2.7:	Jain and Balakrishnan’s bezel menus are activated by a swipe in from the bezel onto the screen from one of eight points (b) and a menu item is selected once the swipe has crossed a threshold distance (a).....	15
Figure 2.8:	In Kurtenbach’s hierarchic marking menus, the shape drawn while navigating the radial menu while a novice (a), is the same used by an expert for the same selection without the menu (b). Hence the expert behaviour is rehearsed when one is still a novice.....	16
Figure 2.9:	A CommandMap for Microsoft Word 2010 that presents all ribbons simultaneously	17
Figure 2.10:	An example performance curve demonstrating the power law of practice for one task performed over time	19
Figure 3.1:	The hidden menu system with all menus open.....	20
Figure 3.2:	(Left) Click interface with all toolbars hidden. The left side of the screen is tapped to make the left toolbar appear (Right).....	21
Figure 3.3:	The toolbar item is selected by tapping on it.....	21
Figure 3.4:	(Left) Swipe interface with all toolbars hidden. The left side of the screen is tapped to make the left toolbar appear (Right).....	22
Figure 3.5:	The toolbar item is selected by swiping towards the bezel next to it	22
Figure 3.6:	Interface for ButtonOpen method.....	25
Figure 3.7:	Interface for new Swipe method.....	26
Figure 3.8:	Graph of average block completion time.....	31
Figure 3.9:	Graph of average errors per block	32

Figure 3.10: Graph of average menu opens per block	33
Figure 4.1: ButtonOpen interface with all toolbars hidden.....	39
Figure 4.2: ButtonOpen interface with left toolbar shown. Selection: tap to open toolbar, then tap item	39
Figure 4.3: Timeout interface with toolbar shown. Selection: user presses and holds anywhere on edge, then taps item when toolbar becomes visible.....	40
Figure 4.4: DoubleTap interface with bottom toolbar shown. For standard selection, user ... taps on edge, then double-taps border at the location of the desired item	41
Figure 4.5: DoubleTap interface showing shortcut selection: user double-taps on the border at item location	42
Figure 4.6: Swipe interface with left toolbar shown. Border colours indicate items boundaries. For standard selection, the user taps edge to open, then swipes item.....	43
Figure 4.7: Swipe interface, shortcut selection: user swipes across item border from inside to the outside	43
Figure 5.1: Mean selection time, by technique and block.....	47
Figure 5.2: Mean incorrect selections per trial, by technique and block	48
Figure 5.3: Distance from menu-invocation gesture to target item (in number of items), by block.....	49
Figure 5.4: Percentage of selections made with the accelerated version of the technique, by technique and block.....	50
Figure 5.5: Mean selection time (Swipe data) by item location (border: actual item location; centre: location categories).....	51
Figure 5.6: Mean distance from menu invocation to target (Swipe data), in items, by item location (border: actual item location; centre: location categories).....	52

CHAPTER 1

1 INTRODUCTION

1.1 INTRODUCTION

Hidden controls are a common user interface technique for making functionality available without taking up valuable space that is needed for application content. Pulldown and popup menus are early examples of the technique, but recently several new forms of hidden toolbars have appeared – e.g., navigation drawers in Android (see Figure 1), the bezel-swipe “charms bar” in Windows 8 (see Figure 2), or the sliding toolbox widgets in Visual Studio. These controls are particularly evident in situations where display space is limited – such as mobile devices that have smaller screens.

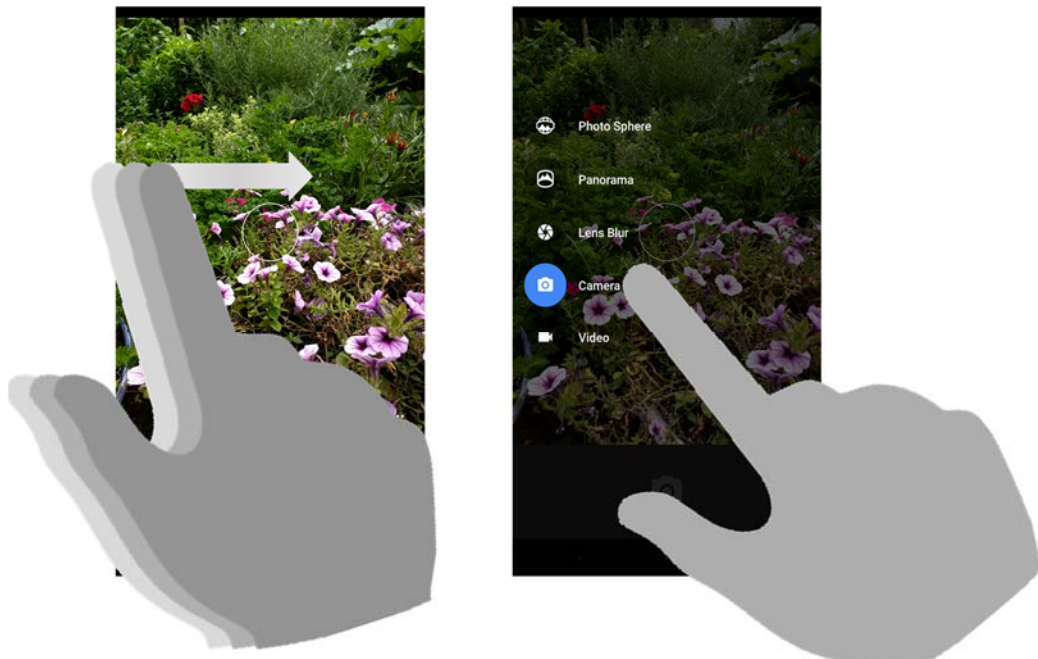


Figure 1.1. Swiping across the bezel edge activates Android’s navigation drawer.



Figure 1.2. Swiping across the bezel edge activates Windows 8’s “charms.”

There are several ways that hidden toolbars can be designed, but most versions require two main types of interaction. First, there must be a mechanism for exposing the toolbar, and second, the user must be able to select their desired item. For example, in the Android navigation drawer, users expose the toolbar by swiping across the device’s left screen bezel and make selections by tapping on an item; in Visual Studio, a button on the screen’s border exposes the toolbar and the user then selects by clicking.

Hidden controls help to maximize display area, but their effects on user performance are less clear – there is little understanding of how best to design these kinds of interfaces to allow both exploration by novices and high performance for experts. In particular, one way in which current designs fail to support expert use is in the two-stage interaction requirements described above. The initial stage of exposing the toolbar ensures that selections can be visually guided – but when people become familiar with the locations of items that they use frequently, they still must carry out both the exposure and selection steps as separate actions. In contrast, accelerated techniques

in other kinds of interfaces (e.g., keyboard shortcuts or command gestures) that do not require such visual guidance for experts can allow for higher performance.

1.2 RESEARCH PROBLEM

Very little research has considered the idea of accelerating selection in hidden toolbars. One recent project demonstrated collapsible toolbars for tabletop use and showed that people can use the collapsed version once they have learned the locations of the toolbar's items [19]. However, there has been no systematic assessment of the design and performance characteristics of hidden-toolbars.

1.3 SOLUTION

To provide this information, we examined design issues and tradeoffs concerning four key aspects of interaction with hidden toolbars: 1) mechanisms for exposing the toolbar; 2) mechanisms for selecting items; 3) spatial memory-based shortcuts; and 4) learnability of different locations. From this analysis, we designed and implemented four hidden toolbars that demonstrate different approaches within this space.

1.4 STEPS IN THE SOLUTION

In order to examine interactions with hidden toolbars, several steps were completed during the research process

1.4.1 Create a framework for hidden toolbar designs

In order to assess hidden toolbars, the first step is to consider what is necessary for the design of the toolbars. Hidden toolbars need to be activated by actions that are distinct from actions used

within the main application. We looked at actions used in pre-existing hidden toolbars such as buttons which are constantly present in the application and bezel interactions. We also wanted to implement accelerated shortcut techniques which used the same interaction to both expose the toolbar and select items from it. This would allow the user to rehearse the behaviour that will be used once they no longer need to expose the menu.

1.4.2 Implement a menu system using hidden toolbars

We developed a mock application with menus on all four sides of the screen, initially hidden, allowing the majority of the screen to be available for the application. Our design framework yielded two mechanisms for item selection: tapping, and swiping; and four mechanisms for exposing the toolbar: tapping the same spot every time (to emulate a standard menu), a held tap, a double-tap, and a swipe. These were combined into four menu systems and two were provided with a memory-based shortcut (double-tap and swipe). The method of tapping the same spot every time cannot inherently have a memory-based shortcut and performing a held tap twice could frustrate users with the time required.

1.4.3 Test and evaluate hidden toolbars interactions in a menu system

To investigate our design, we set up a study where users would be presented with each of the four menus systems in turn and directed to repeatedly select items from the menus within. The items to be selected would be prompted to the user on the middle of the screen. Items would be from a pre-determined set but presented to the user in a random order. This study provided data to evaluate our designs based on criteria such as: selection speed, error rates, if the locations of items are learned over time, etc.

1.5 EVALUATION

We carried out an empirical study to examine the ways that the four designs affected performance (time and errors) and preference. Results largely validated expectations raised through the design space analysis, and they also provided additional insights. Key findings were as follows:

- Hidden toolbar shortcut facilities that exploit spatial memory allow significant performance improvements for experienced users. Importantly, current designs in commercial use do not permit these performance gains.
- Participants quickly started to use shortcut selection methods and made a smooth transition to consistent use (around 90% for both shortcut techniques).
- Switching to the memory-based shortcuts, however, led to error rates that were much higher than the visually-guided techniques (5% compared to ~1%, a statistically significant difference).
- Double-tap and bezel-swiping mechanisms for exposing the toolbar had similar performance (about 700ms faster than the best non-shortcut technique, a statistically significant difference), and participants strongly preferred the swiping technique.

1.6 CONTRIBUTIONS

This work provides three contributions. First, we developed a framework and goals for the design of hidden toolbars. Second, we created four designs exemplifying different approaches and techniques to hidden toolbars. Third, we gained results from a study performed on these toolbars which empirically shows that hidden toolbar shortcut techniques are feasible and desirable on touch devices, and how user learning occurs with them.

1.7 THESIS OUTLINE

Chapter Two will discuss related work and research.

Chapter Three will discuss early studies exploring hidden menus.

Chapter Four will discuss the design of the hidden menus and the research goals behind them.

Chapter Five will discuss the experimental study of the four hidden menus and analyze the results.

Chapter Six will offer answers to the research questions and discuss conclusions, future work, and the contributions of this research.

CHAPTER 2

2 RELATED WORK

2.1 HIDDEN TOOLBARS

Interface design often aims to provide as much functionality to a user as possible. However, this comes with a trade-off: that functionality needs to remain accessible, but not overwhelm the user with options, nor take focus or space away from the main purpose of an application. A common way to handle this is to simply hide the controls until they are required, freeing up valuable screen space. Drop-down hierarchical menus are a common and simple implementation of this idea. The menu itself remains at the top of the application, but it can be clicked on to grant access to a wide range of additional functions whenever a user requires them. That said, drop-down menus have been around for some time and new techniques are currently being explored. For example, navigation drawers in Android, and the charms bar in Windows 8 are both completely absent from the screen initially until a swiping action is performed to expose them. Visual Studio uses sliding toolbox widgets which hide a toolbar while not needed but pressing a perpetually present button allows it to be accessed (see Figure 2.1). Hidden Toolbars are increasingly relevant as mobile devices such as smartphones and tablets become more ubiquitous. With their smaller sizes, maximizing the screen space becomes more important than ever.

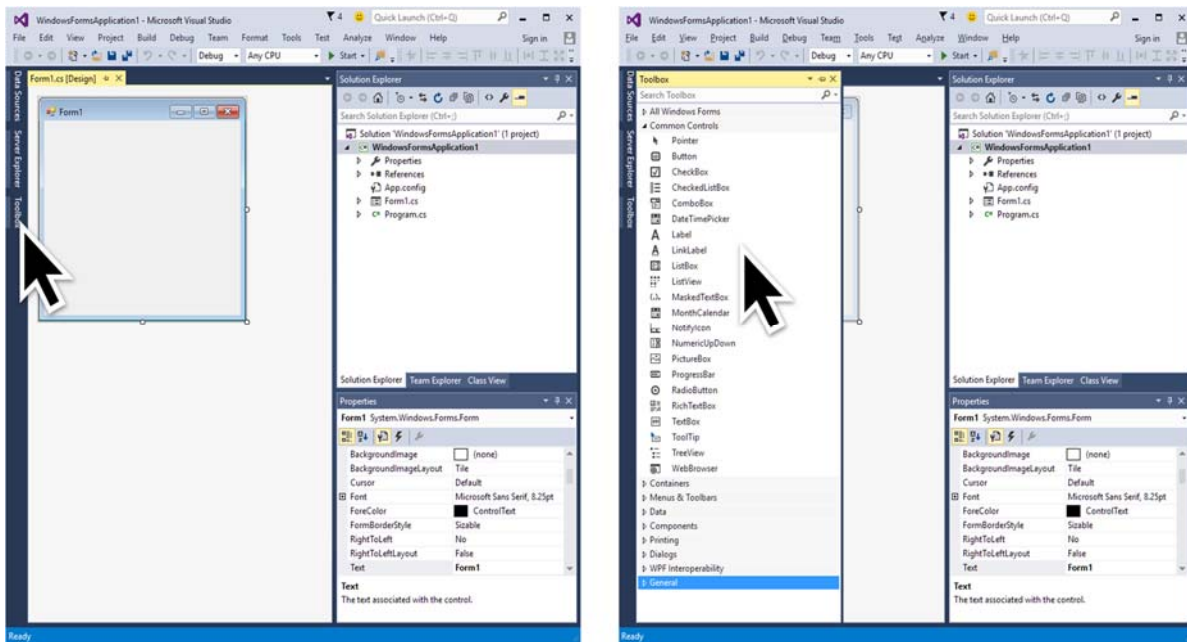


Figure 2.1. Display-edge hidden toolbars in current software systems. Swiping across the bezel edge activates Android’s navigation drawer (top left) and Windows 8’s “charms” (top right). Clicking on a button exposes Visual Studio’s “toolbox” (bottom).

2.2 COMMAND SELECTION

Faster and more convenient methods for command invocation have been extensively examined. Our research on hidden toolbars falls within a broad set of prior research on input techniques that allow rapid and expressive command identification. Examination of touch-based interactions is

particularly prevalent, partially because the finger has different expressive capabilities to those of a mouse (not least, the lack of buttons and a tracking state).

Very briefly, prior interaction techniques have examined a wide range of different parameters to increase input expressivity, including the following (with many hybrids). *Movement direction* was used in Marking Menus [21, 26], in which menus were arranged as segments of a circle centred on the contact point. *Gesture shape* was used in hierarchical marking menus, in which users linked a series of marking menu selections [22] (see Figure 2.8), and it was also used for character entry in systems such as unistrokes [16]. Gesture shapes are now in widespread use for shorthand word entry [44], and recent research has examined how to assist users in learning gestures through dynamic guiding feedback [7]. The *number of touch contacts* [6], their *posture* [34], *pressure* [30, 31] and *rhythm* [15] have also been used to convey information from the user to the system.

2.2.1 Toolbars

Toolbars have long been a standard component of desktop interfaces. They are an extension of the idea of menus in that they combine text with images. This creates a more visual focus than a text-only menu, but also raises the issue of screen real estate since images often require additional space. Early work in the design of toolbars and the use of screen space was done by Au and Li [5] with a widget that allowed toolbars to collapse. Research systems demonstrating and evaluating new forms of toolbars include the *Hotbox* [25], which increased the functional vocabulary accessible through toolbars and allowed the toolbar to be accessed without moving to the display edge, see Figure 2.2. Many of the Hotbox ideas were extended in T3 [24], which used bimanual input to let the user place the toolbar anywhere on the work surface and to coordinate their work via ‘click-through’ tools.

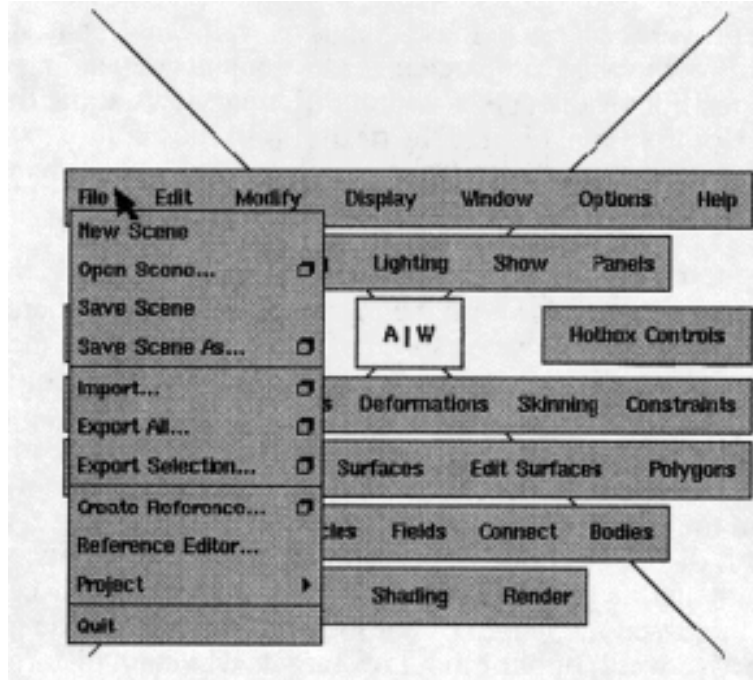


Figure 2.2. A Hotbox widget, as it would appear in the centre of a screen, using a linear drop-down menu [25].

Hinckley *et al.* [18] described an early mobile device that periodically hid interactive controls to maximize display content. CommandMaps [36, 37] extended the range of functionality accessible via a hidden toolbar, making almost all of the functionality for an application available via the hidden toolbar, with a modal transition between workspace view (no commands) and command view (all commands). Finally, Hinckley *et al.* [19] described the use of bezel edge interactions to expose hidden toolbars, see Figure 2.3.

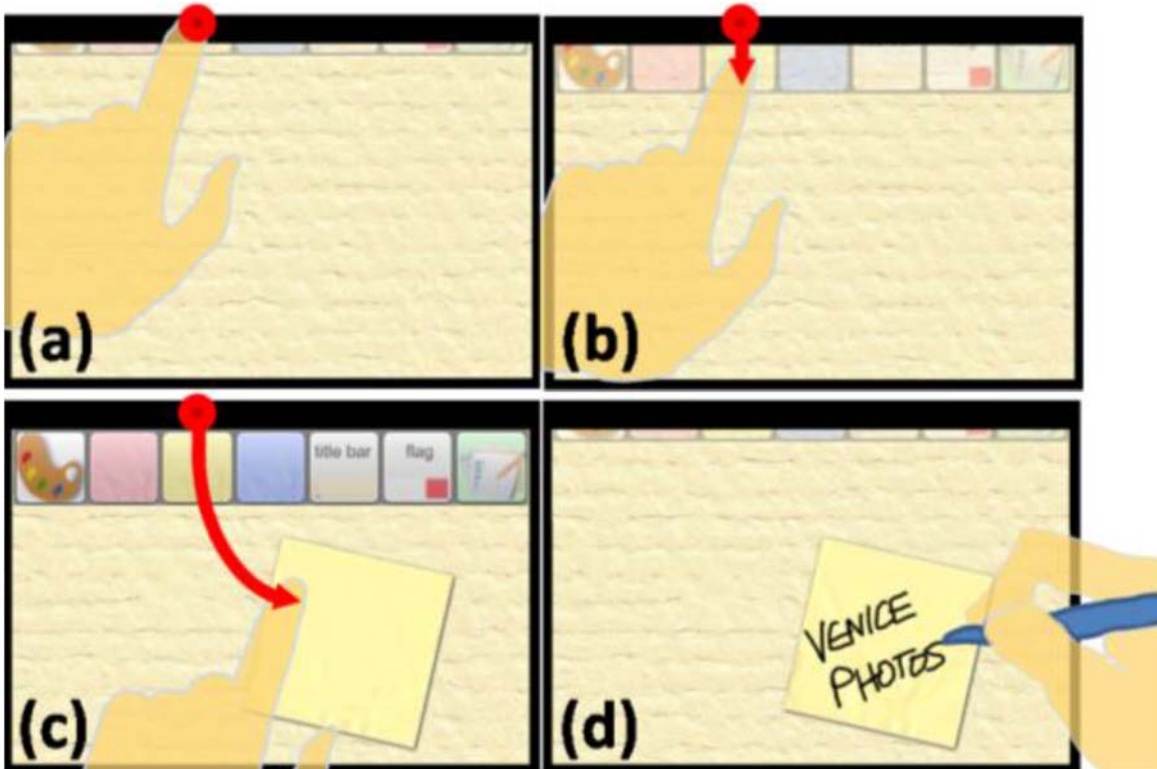


Figure 2.3. Hinckley *et al.* placed a narrow bar at the bezel at the top of the screen (a), when the user swipes down from the menu it expands the bar to a full menu (b). The user can then tap menu items (c) to create items and collapse the menu (d) [19].

2.2.2 Crossing

Crossing techniques [1, 2, 27] have also been proposed to facilitate rapid selections. Tap and click selections require that the finger (or cursor) decelerate to an almost stationary value prior to a tap or click, consuming time. In contrast, sweeping through the target completes a crossing selection, without any need to decelerate the pointing device/limb. Further, the same motion could be used to sweep through more than one target allowing for multiple selections to occur at once. The practical potential of crossing-based interaction was strongly demonstrated by the CrossY [4] drawing tool, see Figure 2.4.

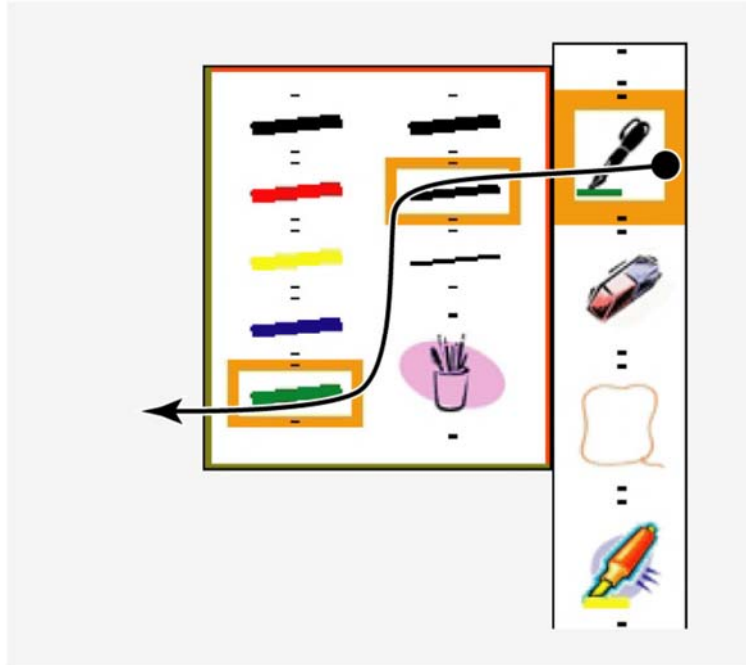


Figure 2.4. The CrossY drawing tool. A single motion selects both the colour and width for a pen stroke with no deceleration required in the targeting [4].

2.2.3 Bezels

Bezel interactions assign special meaning to touches at the display edge or which cross the edge. Hinckley *et al.* [18] described a bezel touch technique in which tilt-based scrolling occurred when the user touched a region near the bezel. Early touch surfaces on mobile devices typically had a raised bezel edge, creating a pointing barrier that facilitated pointing stability, which was particularly beneficial for users with motor control impediments (e.g., [14, 42]). Flush-mounted touch surfaces also created new interaction opportunities for selections that seamlessly transition across the display edge (internally or externally) [8, 33]. Roth and Turner [33], created bezel interactions to design techniques for touch surfaces similar to what could be done with a mouse while not interfering with pre-existing gestures such as pinching to zoom-out the screen. Their Bezel Swipe allowed techniques such as multiple selections, cut, and paste to be possible with gestures, see Figure 2.5.

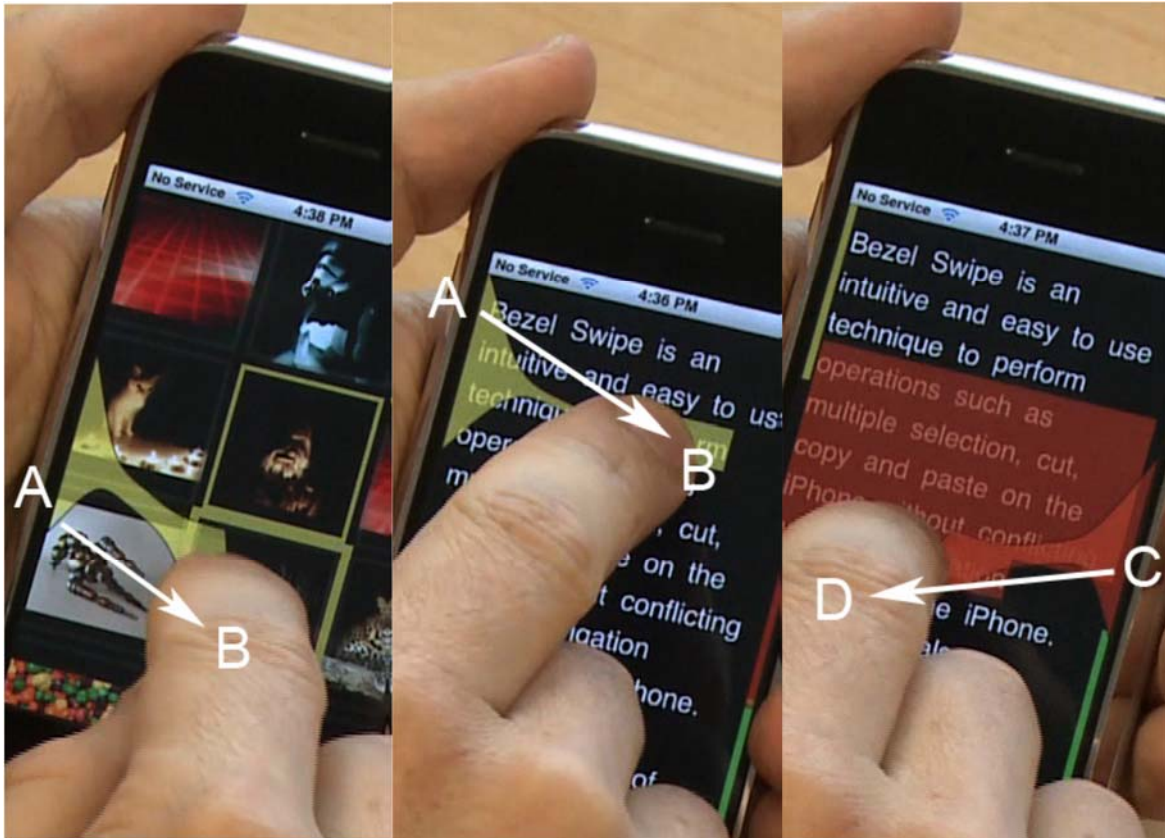


Figure 2.5. Three examples of Bezel Swipe gestures. (Left) A swipe in from the bezel at A to B on the screen selects an image. (Centre) A swipe in from the bezel A to B on the screen starts a text selection. (Right) A swipe in from C on the bezel to D on the screen ends the text selection [33].

Roth and Turner also noted that since it can be difficult for a user to successfully activate a target that needs to be tapped which is right against the edge of a screen the size, a swipe gesture moving into or out from the bezel is much more likely to successfully activate such a target and can therefore make use of screen space that is not typically used in touch interactions. See Figure 2.6.

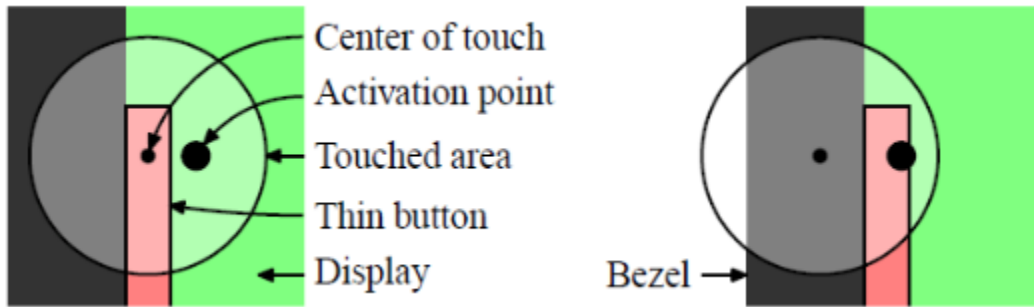


Figure 2.6. (Left) A tapping gesture centred on the target at the edge of the screen may not successfully activate it but a swipe starting with a touch centred on the bezel (Right) is more likely to do so [33].

More recent research on bezel-based interactions has included techniques to assist one-handed thumb selections in display areas that are otherwise hard to reach [43]. Jain and Balakrishnan [20] also did an examination of eyes-free interaction with bezel menus. Typically, a touchscreen device always requires visual attention, especially if one is to make sure that they are selecting a menu item rather than just browsing a menu. Jain and Balakrishnan found that gestures initiated at the bezel could solve this mode-switching problem and the physical form of the device allowed for a combination of landmarks and marking-menu style gestures (discussed further in section 2.3.1) to allow eyes-free interactions, see Figure 2.7. Bezel interactions have also already seen use in a commercial application such as Android and Windows devices, see Figure 2.1.

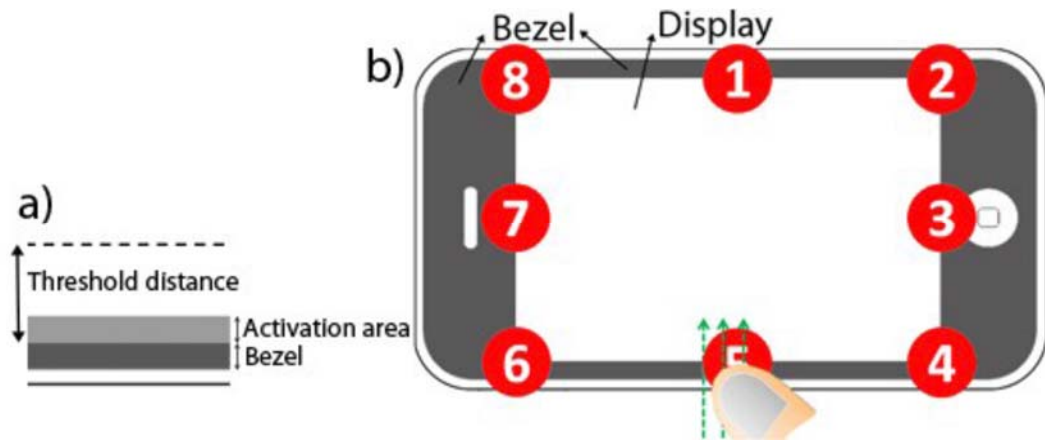


Figure 2.7. Jain and Balakrishnan's bezel menus are activated by a swipe in from the bezel onto the screen from one of eight points (b) and a menu item is selected once the swipe has crossed a threshold distance (a) [20].

2.3 TRANSITIONS TO EXPERTISE

Novice behaviour is the actions used when a user is still learning a system. In a system like Microsoft Word, if a novice user wants to make their text centre-aligned, this would include searching for the menu ribbon the user needs, searching that ribbon for the needed menu item, and finally selecting the item once found. Expert behaviour is the actions of someone who has used the system often over a longer span of time, knows it well, and has possibly even learned shortcut techniques. An expert user of a Microsoft Word menu not only requires less to time to search the menu system for the item for centre-aligned text, but they may also know that they can use the keyboard shortcut Ctrl-E to perform that function without searching the menus at all. Novice and expert behaviour can easily be designed independently of each other but it is more difficult to inform the user of an expert behaviour and transition them into it. In our Microsoft Word example, many keyboard shortcut techniques are not easily found within the program itself and many users often learn of them outside the program from other users who have already learned these expert techniques.

Human factors of expertise development, and interface techniques for supporting transitions to expertise, were recently reviewed by Cockburn *et al.* [10]. Three general strategies for improving performance in command invocation address improvements in the *motor domain* (i.e., muscle memory), in *spatial memory*, and in *methods facilitating memorization*.

2.3.1 Motor domain

A key objective for Kurtenbach's research with marking menus [21-23] was to demonstrate and validate his 'rehearsal principle'. He argued that in order to facilitate transitions to expert performance, a novice's motor actions should be a physical rehearsal of the expert's actions, see Figure 2.8. This principle is contradicted in many interface shortcut facilities (such as keyboard shortcuts) but is supported in marking menus where the same gestural actions are completed when novice and expert, although at different rates and without feedback once expert. Several systems have examined ways to support the rehearsal principle, including ExposeHotkey (for keyboard shortcuts) [29], FastTap (for tablet command selections) [17], and Hinckley *et al.*'s hidden toolbars [19].

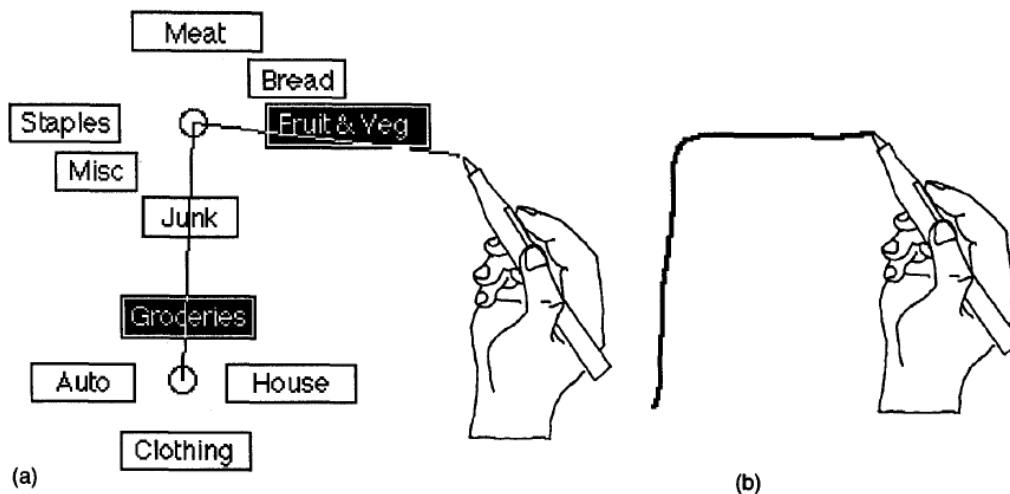


Figure 2.8. In Kurtenbach's hierarchic marking menus, the shape drawn while navigating the radial menu while a novice (a), is the same used by an expert for the same selection without the menu (b). Hence the expert behaviour is rehearsed when one is still a novice [22].

2.3.2 Spatial memory

Novice users typically need to visually search for their intended commands, which is very slow compared with the expert's ability to anticipate their target's location [9]. Several interfaces have been designed to facilitate rapid interaction by explicitly utilizing human spatial memory (e.g., [32, 35-37]). For instance, Scarr et. al. developed a technique called CommandMaps that presents all the commands of an application at the same time in a consistent arrangement allowing a user to learn the location of commands over time, see Figure 2.9. They were able to use CommandMaps to demonstrate that a system that exploited spatial memory was significantly faster than hierarchical menus for expert users while not introducing new obstacles to novice users [36].

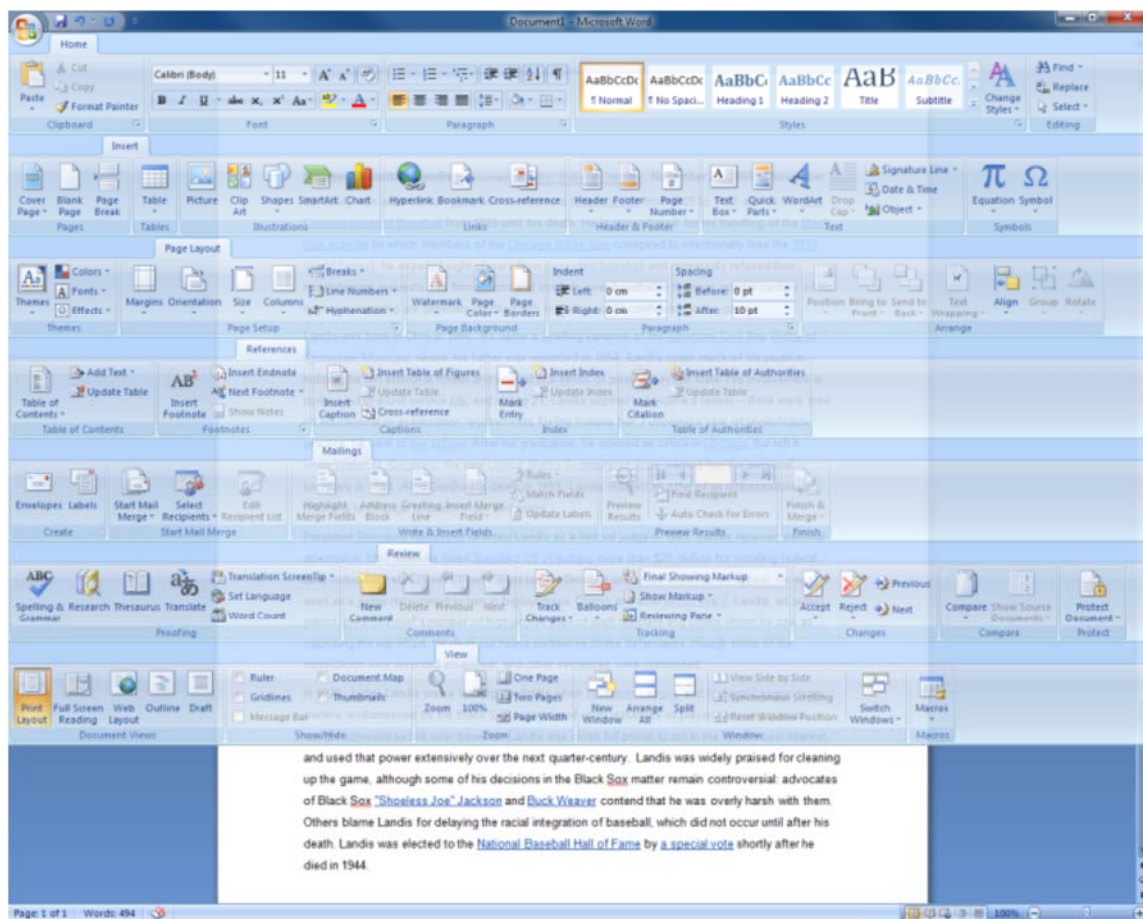


Figure 2.9. A CommandMap for Microsoft Word 2010 that presents all ribbons simultaneously [36].

The proximity compatibility principle (PCP) says that items that are cognitively similar (or proximate) should also be physically similar. Thus, items related to the same task or items with similar functions could be grouped together or share similar icon colours [41]. This principle can be combined with spatial memory. A user may not remember a particular item's location, but if they know the location of a cognitively similar item then they can use that knowledge to look in the same area for the item that they need if the proximity compatibility principle is also utilized.

2.3.3 Memorization

Part of Cockburn *et al.*'s [10] discussion of the transition to expertise is the three-stage model of learning by Fitts and Posner. These three stages are the cognitive stage, the associative stage, and the autonomous stage. The cognitive stage is where one learns the activities that are to be done through direct instruction or observation, often by focusing on the context of the tasks. Performance of tasks in this stage is slow and requires a lot of attention. The associative stage is where one learns how activities are done, without direct instruction. This often involves repeating the task over stretches of time, gaining incremental improvements in performance as the task approaches autonomy. The autonomous stage is when the actions have been rehearsed to such an extent that they can be performed quickly, require so little attention that other cognitive tasks can be performed at the same time, and multiple tasks can often be performed at the same time. When the performance of the task is viewed in aggregate across time, it shows a smooth curve in a continuous power function; with large initial gains in performance that eventually diminishes. This was named the 'power law of practice' by Newell and Rosenbloom, see Figure 2.10.

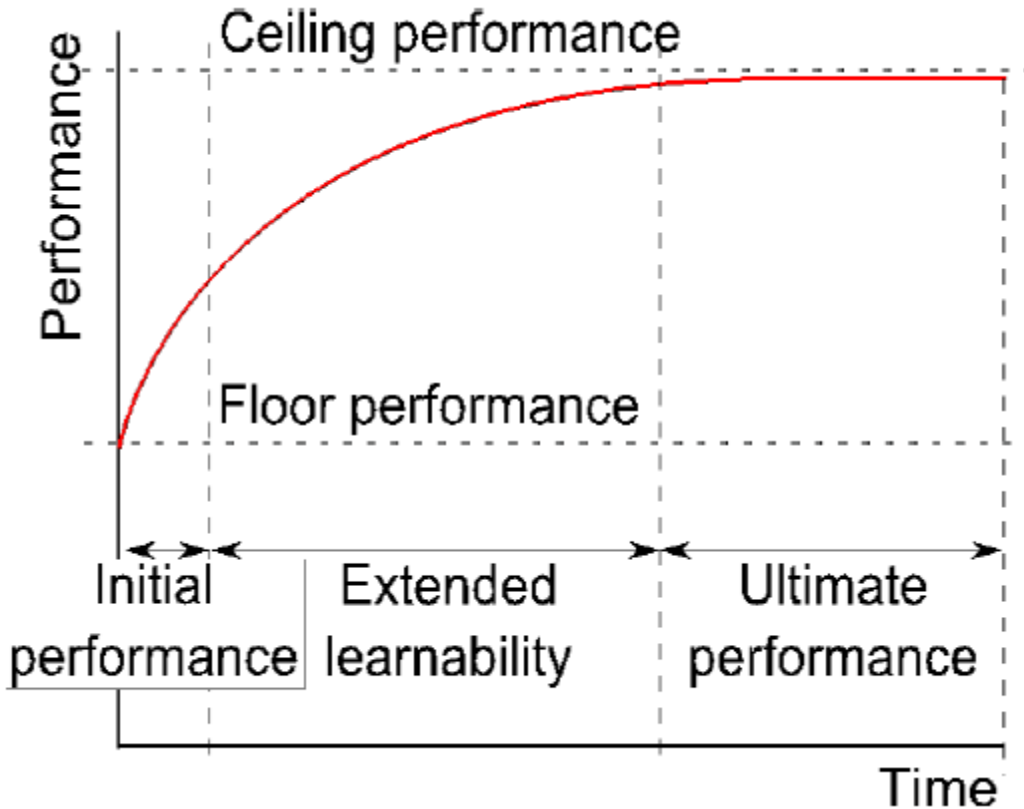


Figure 2.10. An example performance curve demonstrating the power law of practice for one task performed over time [10].

Schmidt's [39] 'guidance hypothesis' suggests that feedback that improves early performance in a task, though assistive guidance, may impair retention of the performed skills once the guidance is removed. In other words, if an interface is too helpful to a novice, users may become reliant on that help, impairing skill development. This finding is reflected in several user interface studies showing that effortful training benefits memorization (e.g., [11, 12]). In a recent study of gestural command invocation, Anderson and Bishof [3] demonstrated memorization benefits when assistive feedback was gradually removed during skill development.

CHAPTER 3

3 EARLY DESIGNS AND PILOT STUDIES

3.1 INITIAL DESIGNS

A preliminary investigation of the hidden menu system was conducted in the summer of 2013. The experimental interface consisted of a blank canvas surrounded by four menus on the edge of the screen. All menus are initially hidden from the user. Figure 3.1 shows the system with all menus open, showing all 28 menu items locations. Menu items were represented by language-independent pictorial icons so that users did not require additional context or understanding of the menu items; they only needed to be able to recognize the items and be able to match them to a prompt. Two methods of interaction were made: the Click method and the Swipe method.

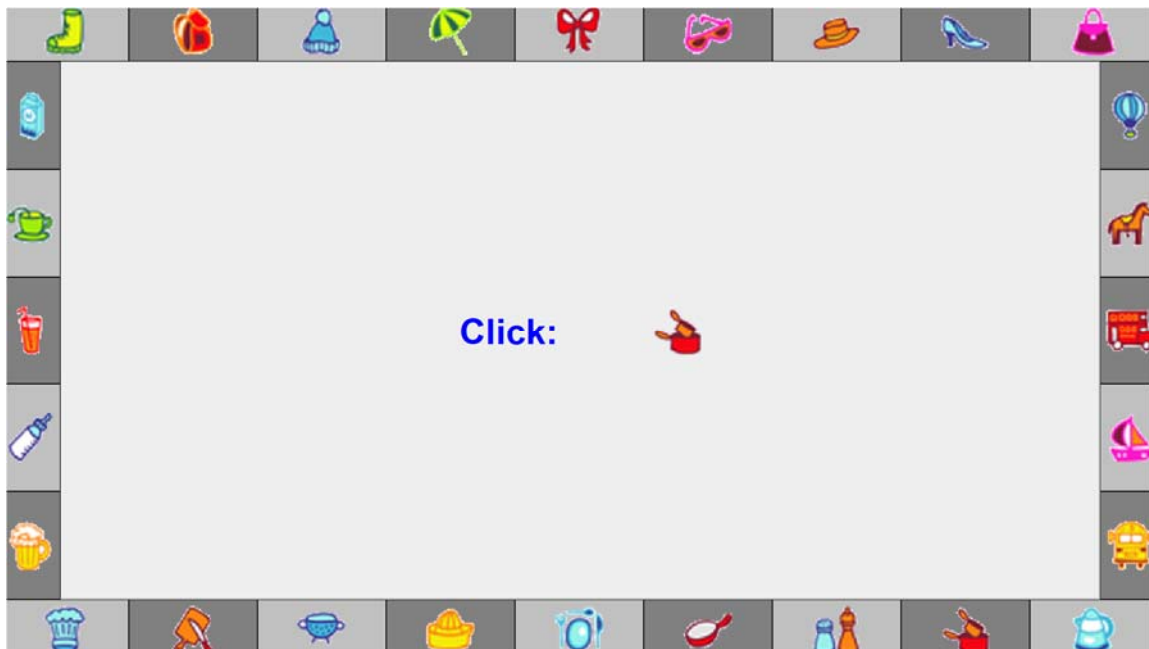


Figure 3.1. The hidden menu system with all menus open.

3.1.1 Click

For the Click method, to make a menu appear, the user must tap near the side of the screen; this will make only the menu on that side of the screen appear, not all four. This is shown in Figure 3.2.

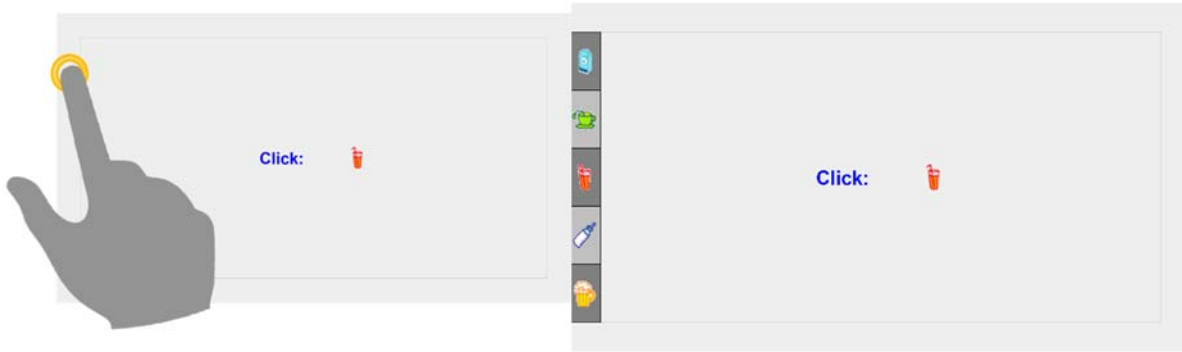


Figure 3.2. (Left) Click interface with all toolbars hidden. The left side of the screen is tapped to make the left toolbar appear (Right).

To select a menu item, the user then taps on the desired item while the menu is open, see Figure 3.3.

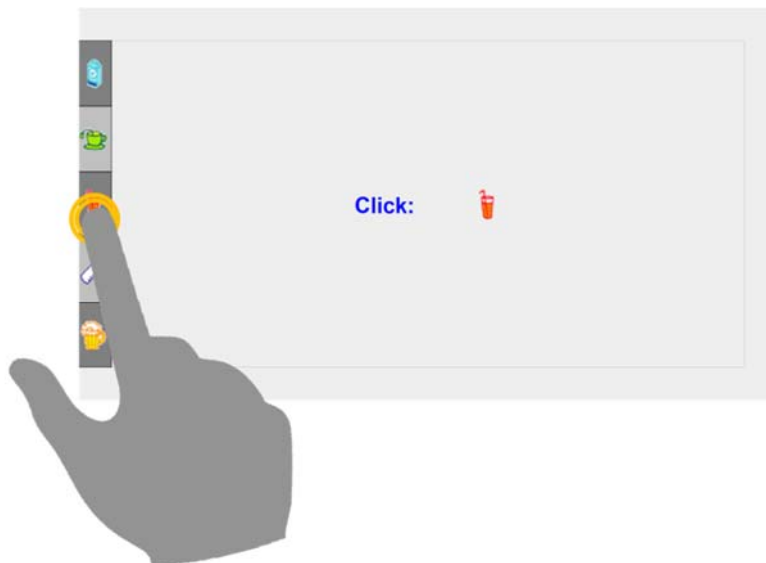


Figure 3.3. The toolbar item is selected by tapping on it.

3.1.2 Swipe

For the Swipe method, menus were opened in the same way: the user must tap near the side of the screen, see Figure 3.4.



Figure 3.4. (Left) Swipe interface with all toolbars hidden. The left side of the screen is tapped to make the left toolbar appear (Right).

For this method, at the very edge of the screen, before the bezel, is a series of thin (7 px) buttons referred to as 'gates,' one for each menu item. To select a menu item, the user swipes their finger along the screen into this 'gate', see Figure 3.5.



Figure 3.5. The toolbar item is selected by swiping towards the bezel next to it.

Once the user begins to use spatial memory to remember where particular menu items are, selection can be performed without first opening the menu by using the same swiping motion.

3.2 INFORMAL EVALUATION

The system was implemented in Java on a Microsoft Surface 2 Pro (10.6-inch screen, 1920×1080 resolution). A study was performed with 12 participants (6 female; mean age 28.8 years). Half of the participants used touch screen devices on a daily basis; the other half of the participants varied between rare and occasional use of touch screen devices. For the study, the system provided the user with a prompt to find a menu item with the shown icon and select it with the prompted selection method. This item was chosen by the system from a pre-determined set of 10 menu items but were presented to the user in a random order (sampling without replacement). Item locations did not change, but each method had separate sets of icons and their stimuli were in unique locations. A successful selection completes a single trial. Any errors made were considered a part of the trial and were included in the total time taken to complete a trial. A set of 10 trials was considered a block. Participants completed 15 blocks for each selection method, with breaks between each block. The starting selection method was swapped between subsequent participants. Participants were free to choose which hand they used but were asked to use the same hand for all trials. There was no set starting position for the participant's hand, nor were they required to return their hand to a particular location between trials.

Observation of the participants showed that most were able to transition to expert behaviour quickly and easily. The data also reflected this, showing that the average trial completion times decreased over time and the average number of times a menu was opened to select an item also decreased over time. However, the data showed that there was not a statistically significant difference between the average trial completion times of the two selection methods. Additionally, the Swipe method had significantly more errors (20% errors in Swipe and 9% errors in Click).

One factor is that participants discovered an unanticipated method of speeding up the Click method by double-clicking near the side of the screen instead of waiting for the menu to open; effectively

creating an expert behaviour for the Click method. We also believe that the high error rate with the Swipe method was affected by a lack of visual feedback when using expert behaviour. When participants selected an incorrect item without opening the menu first, they would often assume the selection had not been registered by the system and would then select the same item again two or three times before opening the menu, seeing that they had been attempting to select the wrong item, and finally proceeding to select the correct item. These extra errors would also contribute to the Swipe trial times. In comparison, the Click method menu would still briefly appear even if a participant was using the double-click behaviour.

3.3 SECOND DESIGNS

A modified hidden menu system was created with four new interaction methods. Four different methods of opening the menus and selecting items from them were implemented in order to determine what method is most efficient. These methods are called ButtonOpen, Timeout, DoubleTap, and Swipe. As before, experimental participants practiced selection tasks with the system and performances were compared. Over time, participants learn and remember where an item is, become more efficient at selecting the item, and learn to use the expert version of selection.

3.3.1 ButtonOpen

The first method, ButtonOpen, is meant to mimic a menu interaction that is currently in use in some applications, such as Visual Studio. The menu is hidden, but each menu has an additional button which is always visible to the user. This button controls whether the corresponding menu is hidden or visible. While the menu is visible, an item can be selected, whereupon the menu disappears again. In this method, there is almost no difference between novice and expert behaviour. This double-selection action, of first the menu-opening button followed by the desired menu item, as well as visual search for the desired menu item are required regardless of user experience. The only difference could be that a more experienced user might be able to perform the item search a little more quickly if they already know the location of the desired menu item. See Figure 3.6.



Figure 3.6. Interface for ButtonOpen method.

3.3.2 Timeout

The second method, Timeout, requires the user to tap near the side of the screen and then continuing holding their finger on the screen for a short while in order for the menu to appear, similar to the behaviour used for marking menus, described previously. Once the menu is visible, the desired menu item may be selected. While this method still requires two actions in both novice and expert mode, the user can now open the menu by pressing anywhere on the menu, not just at a single stationary button. This would allow the user to make use of spatial memory to remember where a menu item is and to open the menu near that point which would allow them to prevent some of the searching for the item. The use of the pause, or ‘time-out’, to make the menu appear is deliberately meant to slow the user down and prevent the double-click behaviour seen in the preliminary investigation. This is intentionally done so that the two behaviours may be investigated separately.

3.3.3 DoubleTap

The third method, DoubleTap, adds a small border around the edge of the screen. This border is divided into coloured sections representing the hidden menu items (the colours match the

predominant colour of the menu item icon). To select an item here, the user can tap the screen on just before the coloured border to make the menu appear and then double-tap within the coloured border for the desired menu item. For expert behaviour, the user can double-tap these (15 px) coloured sections without making the menu appear first. The colours are intended only as a hint to aid the user in remembering where a particular menu item is and were not considered a factor in the learnability of the menu items, nor were they a part of the analysis.

3.3.4 Swipe

The final method, Swipe, keeps the coloured border from the previous method. This method is similar to the initial Swipe method from the preliminary investigation. As before, in the DoubleTap method, to make the hidden menu appear, the user taps their finger near the edge of the screen just before the coloured border. To select an item, the user performs a swiping action towards the coloured border and the edge of the screen at the desired menu item. The (7 px) ‘gates’ are now the pieces of the coloured border itself. The expert behaviour uses the same swiping gesture for item selection but the user does not need to have the menu visible to select an item. See Figure 3.7.



Figure 3.7. Interface for new Swipe method.

In addition to the colours, another method to aid the user's memory is also employed here: whenever a selection is made, the item selected will flash briefly on the screen. This is intended to help prevent user confusion if the wrong menu item is selected. Instead of seeing nothing happen and assuming that the selection was not registered by the system, if the icon of the selected menu item briefly flashing on the screen, the user will see that the selection was registered and will be able to tell if they selected correctly or not.

For each interaction method, data logged by the system will include: how long it takes for a correct selection to occur after the prompt is shown, how many incorrect selections are made, how far an incorrect selection was from the target based on Manhattan distance, how many times a menu was opened, whether a selection was made while the menu was visible or not, and the distance between the item location and where a menu was opened from. In this way, we can determine which selection method is the most efficient, leads to the least errors, and requires the least amount of time for a user to transition from novice to expert behaviour.

3.4 PILOT STUDY OF SECOND DESIGNS

3.4.1 Methods

The new study was run on a Microsoft Surface 2 Pro tablet (1920x1080 resolution) with eight participants (5 female; mean age 39.4 years). Half of the participants used a touch screen device every day while the other half used touch screen devices rarely, if at all. The system provided the user with a prompt to find a menu item with the shown icon and select it with the prompted selection method. This item was chosen by the system from a pre-determined set of 10 menu items but were presented to the user in a random order (sampling without replacement). Item locations did not change, but each method had separate sets of icons and their stimuli were in unique locations. A successful selection completes a single trial. Any errors made were considered a part of the trial and were included in the total time taken to complete a trial. A set of 10 trials was considered a block. Participants completed 11 blocks for each selection method, with breaks between each block. Interface order was counterbalanced using a Latin Square design. Participants

were free to choose which hand they used but were asked to use the same hand for all trials. There was no set starting position for the participant's hand, nor were they required to return their hand to a particular location between trials. The experimental data was analyzed to determine how long it takes users to complete the same type of selection with each selection method, whether an item is selected using novice or expert behaviour, and how many errors are made while trying to select the item. The ideal system would have the least time taken to select an item, a short transition period between novice and expert behaviour, and a minimum number of selection errors.

The ButtonOpen method, which has users select a button in the same spot every time they want to open a menu, was expected to be the slowest and least efficient method since it requires moving towards the same target every time even if the user was initially close to the item to be selected, and because it requires the user to search a menu for the item to be selected every time without making use of spatial memory. However, for similar reasons, this method is also likely to create the fewest errors.

The Swipe method, which makes a menu appear by tapping near the edge of the screen and selection is done by swiping towards the edge of the screen, was expected to be the most efficient system. The actions required are simple and they make use of Fitts' law [13] and the infinite width target that the edge of a screen makes. However, since this technique is something which most users are likely to be unfamiliar with, it is possible that this method may involve more errors or could require more time for a user to be comfortable using the expert behaviour. The Swipe method has been designed with the intent of assisting users with these potential issues by adding measures such as showing the item that was selected and displaying colours around the edge of the screen to give a hint as to which item is where. Where appropriate, these techniques have also been added to other selection systems to help avoid an unfair advantage for the fourth method.

3.4.2 Results

We carried out an ANOVA to compare the four designs in terms of completion time, error rate, and the learning curve. ANOVA showed no significant main effect of method on average selection time ($F_{3,21} = 2.16$, $p = 0.12$). There was a significant effect of block ($F_{10,70} = 18.10$, $p < 0.000$),

and a significant interaction between method and block ($F_{30,210} = 3.01, p < 0.000$). It was also noted that the DoubleTap method had a very high standard deviation of 2536.08 compared to the other methods, see Table 3.1. This was corroborated by a paired sample t-test between the DoubleTap and Timeout methods which gave $p = 0.572$, compared to $p = 0.039$ between DoubleTap and ButtonOpen and $p < 0.000$ for all other pairs.

Table 3.1. All trials.

Method	N (num. trials)	Mean (ms)	Std. Dev.	Std. Err.
ButtonOpen	88	2832.23	814.81	38.86
DoubleTap	88	3327.57	2536.08	270.35
Timeout	88	3195.39	614.75	65.53
Swipe	88	2074.29	748.72	79.8

Further investigation showed that one of the participants ran through the DoubleTap method and produced data that fell outside of two standard deviations for all blocks. When these outliers were removed, the standard deviations were more balanced across methods, see Table 3.2. This removed a total of 14 out of 352 data points. Another ANOVA was run with all outliers beyond two standard deviations removed. The new ANOVA did show a significant main effect of the method on average selection time ($F_{3,2} = 19.04, p < 0.000$). There was also a significant effect of trial block ($F_{10,71} = 24.76, p < 0.000$), and a significant interaction between method and block ($F_{30,197} = 3.92, p < 0.000$). Paired sample t-tests between each pair of methods showed that there were statistically significant differences between each method with $p = 0.009$ for ButtonOpen and DoubleTap and $p < 0.000$ for all other pairs.

Table 3.2. Trials after removing outliers.

Method	N (num. trials)	Mean (ms)	Std. Dev.	Std. Err.
ButtonOpen	87	2784.73	686.16	73.56
DoubleTap	76	2426.1	986.9	113.21
Timeout	87	3161.06	526.68	56.47
Swipe	88	2074.29	748.72	79.4

3.5 INTERPRETATION AND OBSERVATIONS

Examination of the test data shows that the Timeout method has the most consistent selection times with the least amount of change over time. ButtonOpen and DoubleTap both start with high selection times that decrease with use through the testing period. These methods have very similar curves and values. The Swipe method starts with the lowest selection times and remains the fastest throughout the test period. See Figure 3.8.

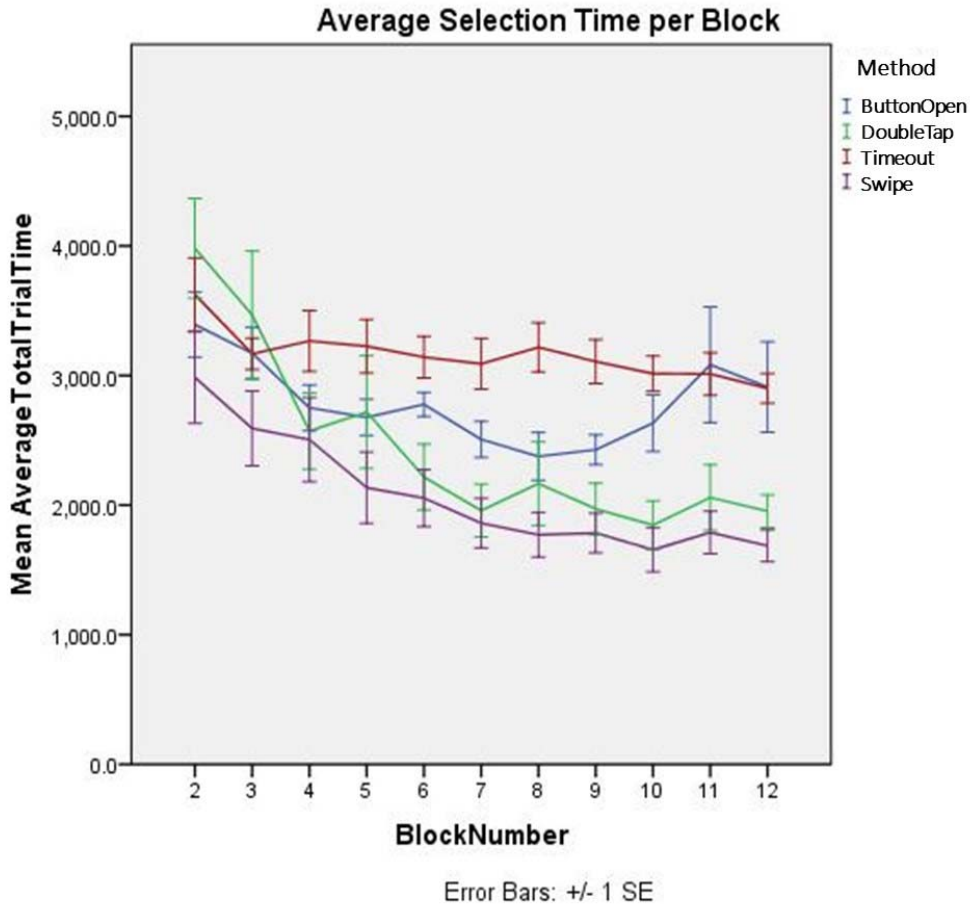


Figure 3.8. Graph of average block completion time.

DoubleTap was the method with the highest number of errors and many participants expressed frustration in using this method, see Figure 3.9. Timeout and ButtonOpen both had almost no errors, which is understandable given that users did not have to guess at the location of menu items with these methods. Errors did occur in use of the Swipe method but there were less than occurred with DoubleTap and the frequency of errors when using Swipe drastically reduced over time. The previous problem of users making the same incorrect selection multiples times almost completely disappeared and was only observed to occur twice. Users stated that the colours used in the Swipe and DoubleTap methods helped them to remember the locations of menu items. This is especially impressive given that users were not told about the association between icon colours and the

coloured borders prior to starting the test. Users were able to determine this on their own and could be observed making use of this even in the first couple of test blocks.

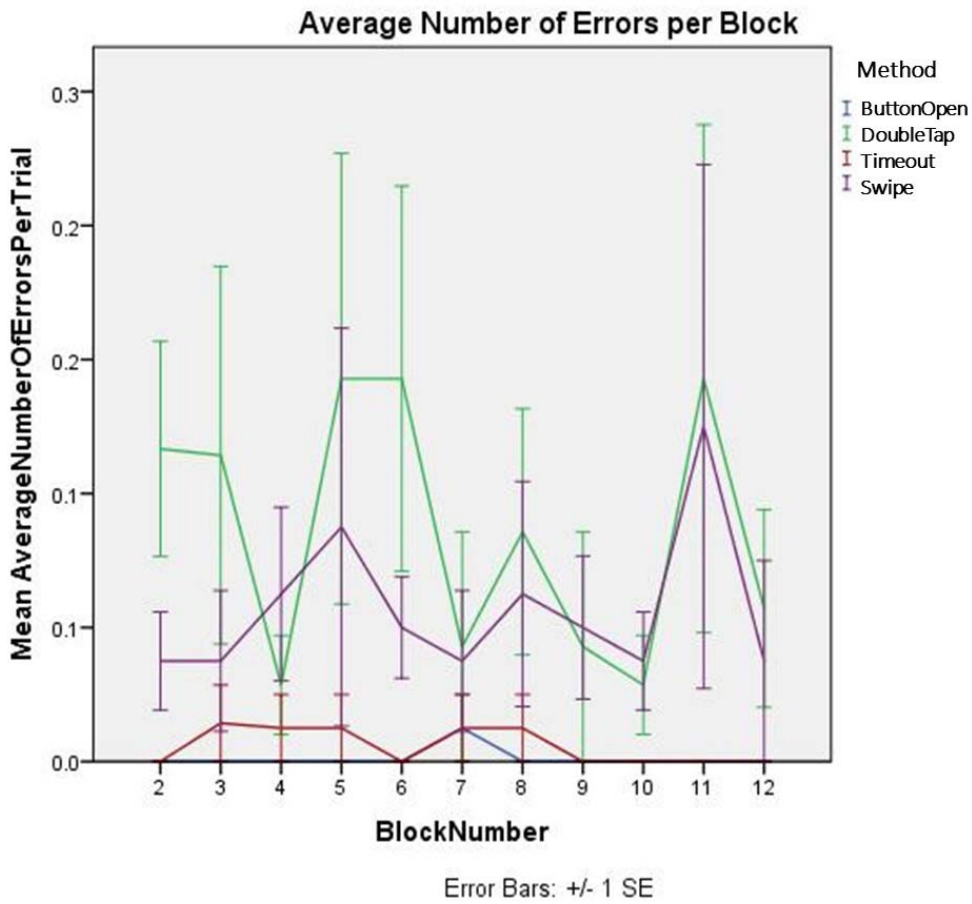


Figure 3.9. Graph of average errors per block.

DoubleTap and Swipe were the two methods that provided an expert behaviour for selection for the user. The decreasing time needed to select menu items, see Figure 3.8, along with the decreasing number of times users would open a menu, see Figure 3.10, show that users made use of this expert behaviour and that it improved their performance. The Swipe method, in particular, shows a smooth learning curve with the number of menus opens approaching zero by the end of the test. Users were observed to start making use of the expert behaviour for Swipe, unprompted, within the first two blocks.

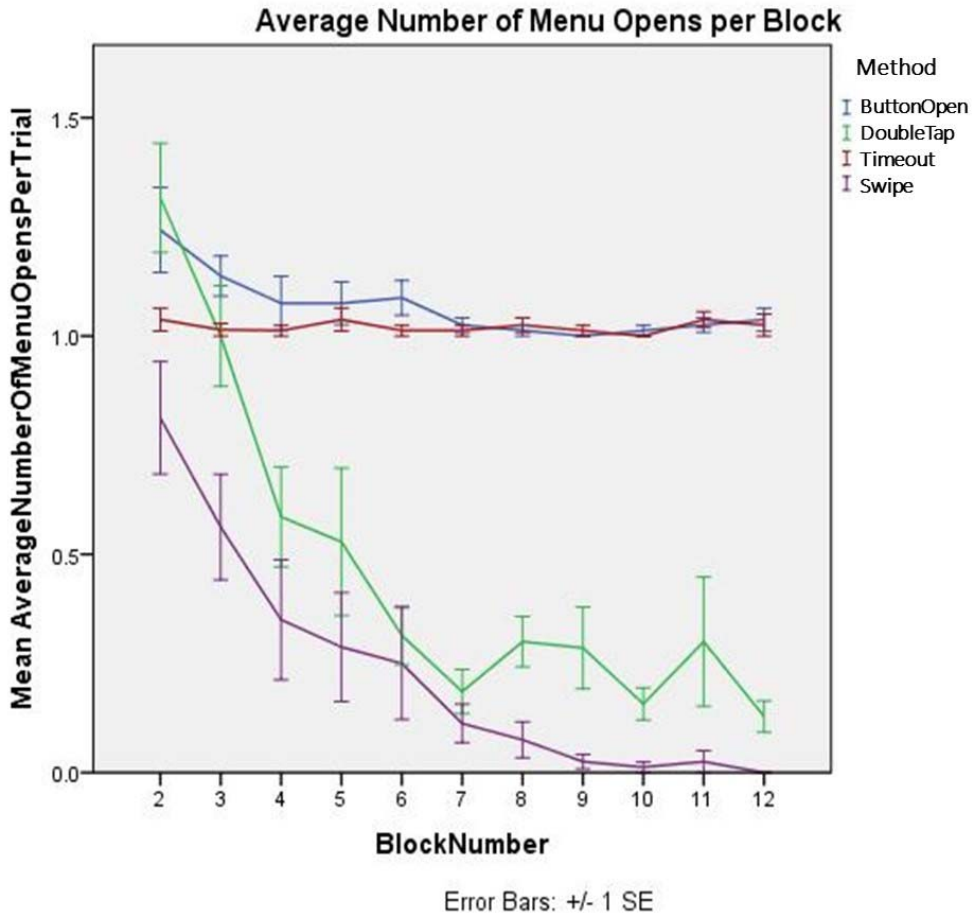


Figure 3.10. Graph of average menu opens per block.

While the DoubleTap method shows a similar trend, it has consistently higher numbers. This is likely to due to some issues that arose in targeting. Users were observed to have difficulty in selecting only the coloured bar around the edge of the screen, their finger often overlapping with the menu area, which caused the menu to open and the item remained unselected. Users also started trying the expert behaviour selection of DoubleTap early on in the test, but it took longer for them to become comfortable in its use and two users needed to be reminded of how to perform the expert behaviour.

One user comment characterizes the DoubleTap method difficulties by saying “[the] small square needing the double tap [is] hard to use, very frustrating.” Users generally found the DoubleTap method to be the most frustrating method to use and the easiest to make errors with. In comparison,

the Swipe method was considered to not be frustrating, led to some of the least errors, and helped the users succeed in their task. Seven out of the eight participants said that Swipe was their preferred method overall out of the four.

Based on the information gathered from these informal design iterations and evaluations, we developed a new framework and designs. We primarily wanted to perform a more robust study with more participants and ensure an even randomness in the order in which the selection mechanisms were presented to them. We also took the opportunity to adjust the DoubleTap method based on participant feedback by enlarging the area the expert behaviour selection was performed on in the hope of reducing user error. These designs and framework are further explored in chapter four.

CHAPTER 4

4 A DESIGN FRAMEWORK FOR HIDDEN TOOLBARS

Based on the understanding gained from the early designs and pilot studies (described in chapter 3), we developed a design framework for hidden toolbars and built four final interfaces based on this framework.

4.1 DESIGN GOALS

Many forms of hidden windows/toolbars are supported across different form factors and applications for a variety of different purposes. Our work is focused on the specific activity of gaining rapid access to common functionality. The four main design goals of hidden toolbars within this project are therefore as follows:

Maximize the available work area. The command invocation interface needs to be predominantly hidden during the interaction to allow the full display space to be dedicated to content (particularly important for mobile devices). Even when it is exposed, it is desirable that the toolbar consume only a small region of the display space, preferably at the display edge, to preserve as much of the screen space as possible for the application content.

Allow easy use for novices. Novices should receive visual feedback that indicates: a) the means for displaying the hidden toolbar; and b) the alternative commands available once displayed.

Allow shortcut mechanisms for users. Experts should be able to quickly select items in a single shortcut action that combines elements a and b above.

Allow access to a moderately large command vocabulary. The interface should allow rapid access to a command set that covers the functionality of most mobile applications; our informal survey of these systems suggests that 30 different commands are a reasonable estimate for this size.

4.2 HIDDEN TOOLBAR DESIGN CONSIDERATIONS

To satisfy the design goals, hidden toolbar interfaces must support mechanisms for (i) affording the availability of the toolbar, (ii) exposing the toolbar, and (iii) selecting (or cancelling) selection of an item within it. They must also support (iv) a shortcut selection mechanism that combines the exposure and selection steps into a single action. Finally, there are potential design considerations around (v) the learnability of items at different spatial locations. Each of these issues is briefly discussed below.

4.2.1 Mechanisms for indicating toolbar availability

There are design tensions between the need to indicate to the user that hidden toolbars are available and the desire to leave the work surface uncluttered with superfluous controls that might occlude content. As edge-based interactions become increasingly common (e.g., the prevalent use of bezel edge swipes to activate a smartphone notification centre) users are likely to increasingly expect their availability and explore to discover them. However, in contemporary devices, it would be unrealistic to expect novice users to guess at the availability of hidden commands, or to guess the actions used to access them. Our designs, described below, demonstrate various degrees of affordance for toolbar availability.

4.2.2 Mechanisms for exposing the toolbar

The mechanisms for exposing the toolbar raise design tradeoffs that are related to those associated with affordances. There are potential conflicts arising from accidental activations – the user may intend to interact with the work surface and instead invoke the hidden toolbar by mistake, or the inverse. We see two main categories of design approaches to reducing this problem, broadly encapsulated as *targeting* and *intentionality*. *Targeting* approaches reduce the likelihood of accidental target acquisition by diminishing the size of the hidden toolbar exposure control, or by requiring that the targeting action includes components that are unlikely to occur in normal interaction, such as requiring a bezel swipe across the display edge (e.g., smartphone notification centres). *Intentionality* approaches require that a special action is conducted to expose the hidden toolbar, where that action has no pre-assigned meaning in the open application. Examples of

specifically intentional actions include a high force touch event, a dwell action, a double tap, or a double-finger tap. *Hybrid* approaches that combine elements of targeting and intentionality are also possible and may offer opportunities for affording the action requirements to complete the action for novice users (e.g., a small balloon icon that squashes when touched and pops when pressed sufficiently hard to expose the toolbar).

4.2.3 Mechanisms for selecting (or cancelling) item selection

Once the hidden toolbar is exposed, a command within it is likely best selected through normal direct manipulation means (pointing and clicking on the target or tapping it). All of our toolbar designs automatically hide the toolbar after each selection, which raises questions about the provision of feedback to confirm the item selected. Our designs use a short duration of visual persistence for the selected item – the hidden toolbar disappears immediately after selection, but the selected item remains displayed for 200ms.

There also needs to be a mechanism for the user to cancel a selection and hide the toolbar when they find that the target item is not present. Our designs dismiss the toolbar on any user action outside the toolbar region.

4.2.4 Shortcut selection

Once users have learned item locations they should be able to use this knowledge to substantially improve their performance, ideally combining the exposure and selection actions into a single rapid action. Furthermore, the *rehearsal hypothesis* [21] suggests that transitions to expert performance will be best supported if the novice's two-step actions serve as a *physical rehearsal* of the actions required when expert. Our hidden toolbar designs support expertise development in two ways. First, they use spatially stable command locations, allowing users to draw on their spatial knowledge to anticipate where target commands can be activated (rather than rely on visual feedback to guide selections). Second, some of the designs allow shortcut selections that use only the second action – i.e., selections can be completed without first exposing the toolbar.

4.2.5 Learnability of items

Finally, there are considerable design issues around factors that make locations and actions easier to learn. For example, it is likely that commands at certain locations will be easier to learn and memorize than others (e.g., display corners may be easier than arbitrary locations along an edge). There are also complex tradeoffs around the use of facets such as visual demarcations along the display edge to assist recall of command locations – while visual clues may assist users at an intermediary level of expertise, there are research findings suggesting that continual visual guidance can impair expertise development because it encourages reliance on the guidance, impeding full memorization [3, 12]. Again, our designs below are intended to probe different points within this design space.

4.3 FINAL HIDDEN TOOLBAR DESIGNS

To explore the design space above, we designed and evaluated four different hidden toolbar interfaces.

4.3.1 ButtonOpen

ButtonOpen is based on the interaction method used in desktop applications such as Visual Studio. The toolbar is exposed when the user taps/clicks on a continually visible invocation button that is displayed in the window border (Figure 4.1). Once exposed, tapping an item completes the selection (Figure 4.2), and tapping on the work surface dismisses the toolbar without a selection.

Experience-based performance improvement is supported in two ways. First, the user will learn which toolbar contains the command (top, bottom, left, or right). Second, users will learn command locations within the toolbars. However, even when the user has memorized the exact combination of toolbar and item location, they must still carry out both the exposure and selection actions.

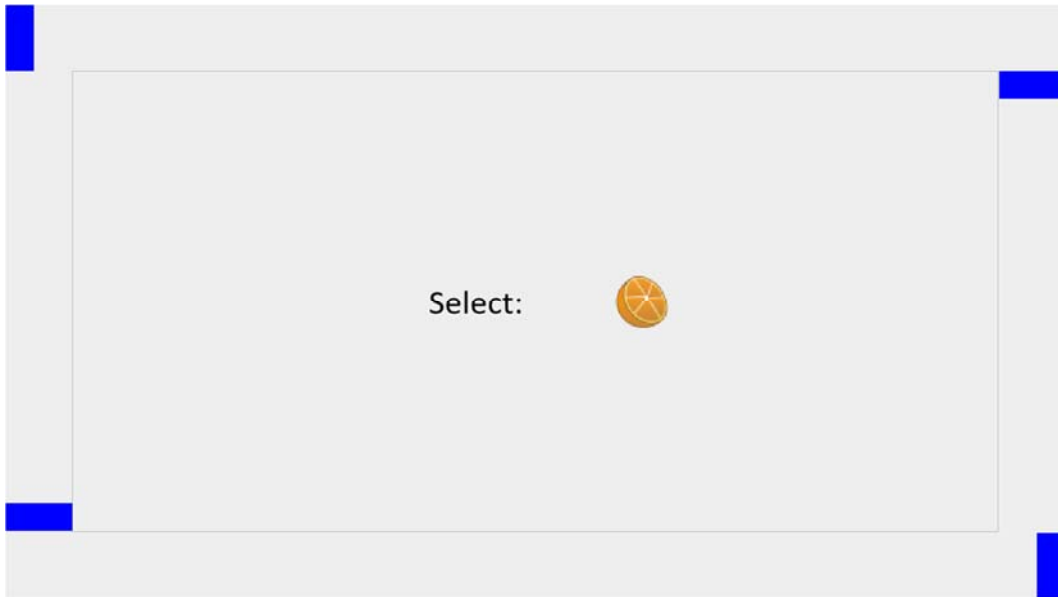


Figure 4.1. ButtonOpen interface with all toolbars hidden.

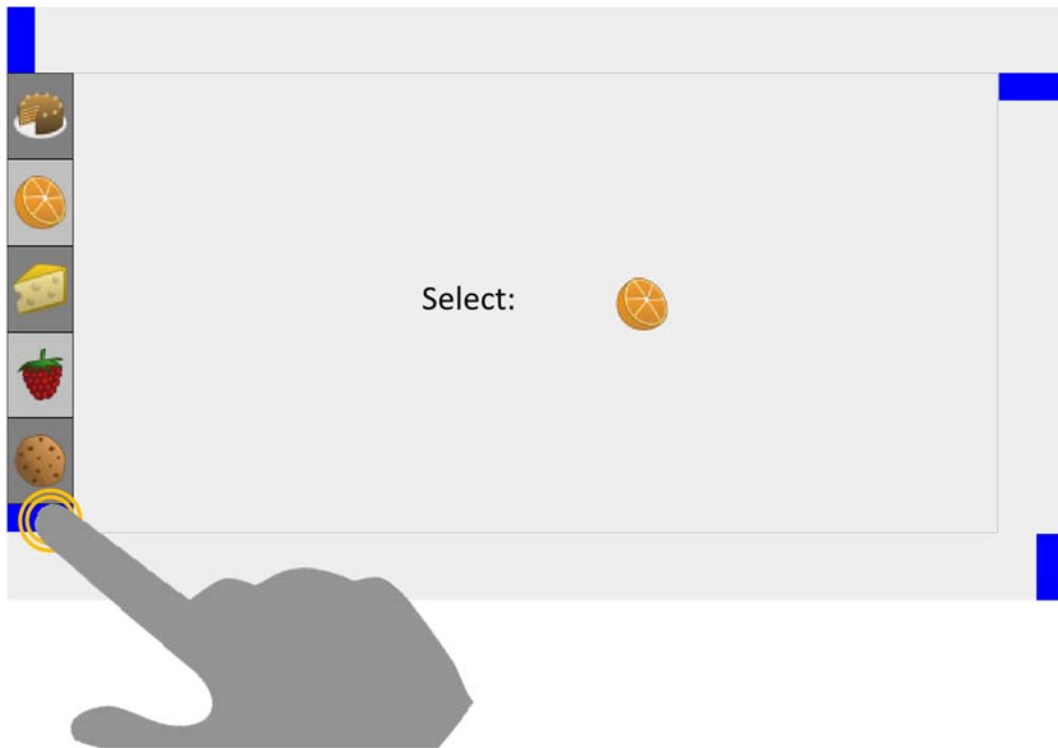


Figure 4.2. ButtonOpen interface with left toolbar shown. Selection: tap to open toolbar, then tap the item.

4.3.2 Timeout

The Timeout method requires the user to touch and hold on the border of the screen for a period of time (475ms in our study system). This timeout provides the invocation mechanism for showing the toolbar. Once the toolbar is shown, the user selects an item by tapping on it (Figure 4.3).

Experience-based performance improvement is supported in two ways. First, as with the ButtonOpen method, users will learn the correspondence between toolbars and their contained items. Second, as users learn the locations of items within the toolbar, they can carry out the press-and-hold invocation gesture nearer to the correct location, reducing the movement distance to the target when it appears. This design, therefore, replaces some of the *targeting* requirements with *intentionality* from our design space, reducing or eliminating the secondary targeting action once experienced. Furthermore, other intentionality triggers, such as force-touch, could be used to reduce the time required to express the user's intention (eliminating the need for a timeout).

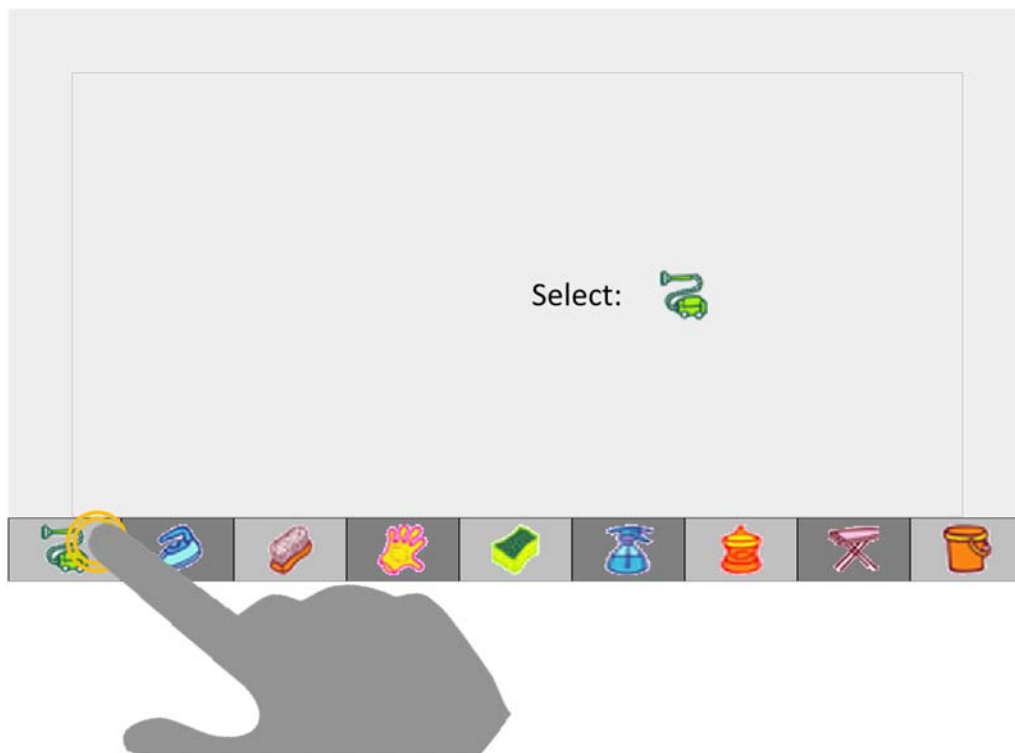


Figure 4.3. Timeout interface with toolbar shown. Selection: user presses and holds anywhere on edge, then taps item when toolbar becomes visible.

4.3.3 DoubleTap

The DoubleTap method uses a 30-pixel-wide visible border with differently-coloured sections to indicate the location of each item in the toolbar (Figure 4.4). DoubleTap provides two ways for users to make selections. When users are unfamiliar with item locations, they expose the toolbar by tapping anywhere on the screen border. A double-tap on an item completes the visually guided selection.

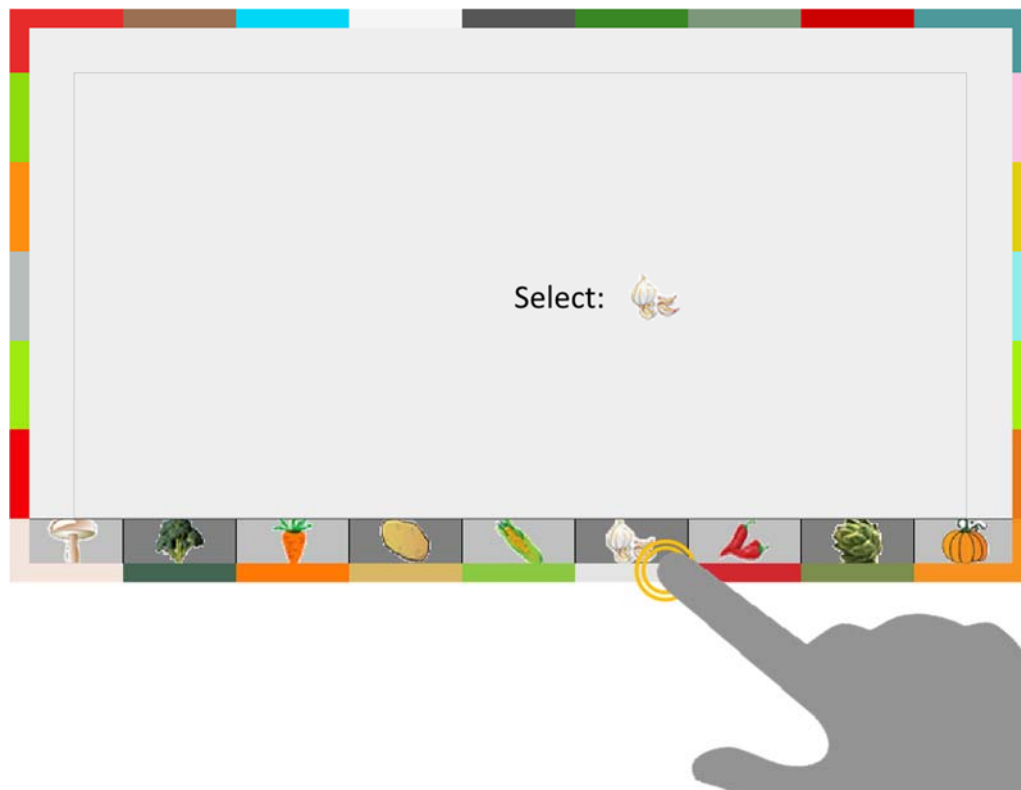


Figure 4.4. DoubleTap interface with bottom toolbar shown. For standard selection, user taps on edge, then double-taps border at the location of the desired item.

Once users have learned item locations, they need only carry out the double-tap action at the location of the intended target (on the coloured border). The selected item is briefly shown after selection (Figure 4.5), but the full toolbar remains hidden to reduce occlusion. Although the coloured regions on the display edge are narrow to maximize workspace area, the targeting region

for a combined double-tap action is 30 pixels wide to assist targeting (double the size of the targeting region in the DoubleTap method in section 3.3.3).

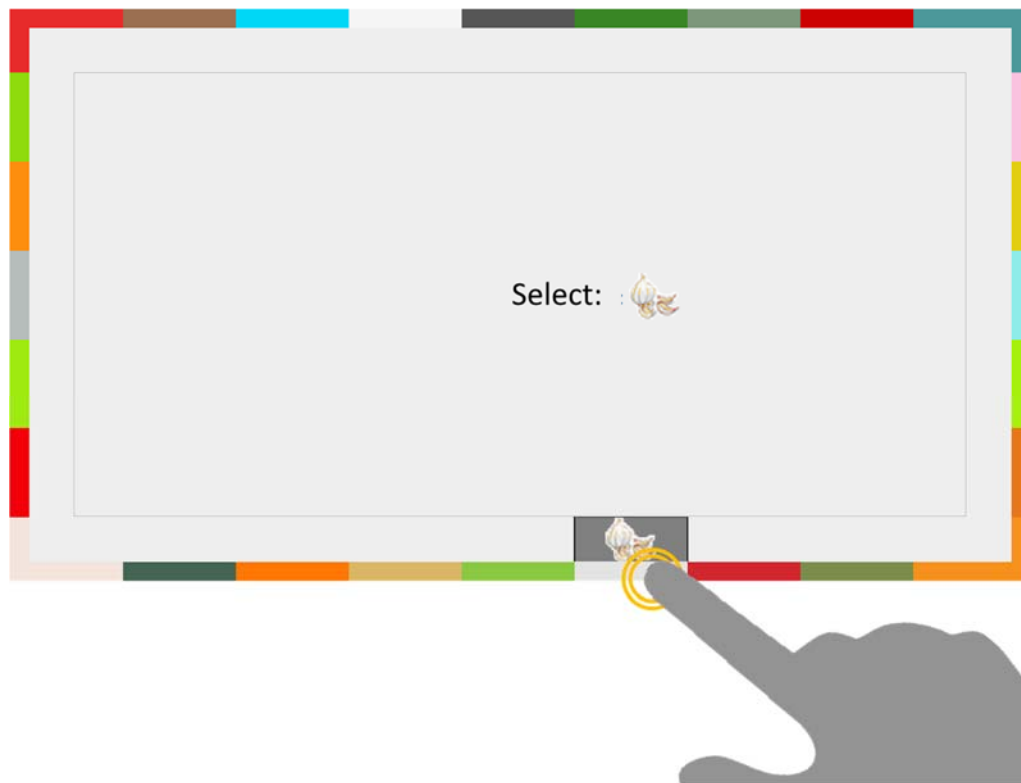


Figure 4.5. DoubleTap interface showing shortcut selection: user double-taps on the border at item location.

4.3.4 Swipe

The Swipe method is similar to DoubleTap but uses gate-crossing swiping actions [2, 4] to complete selections instead of taps. It uses coloured regions to demarcate item locations, but these markers are narrower than DoubleTap (only 7 pixels wide, the minimum that system required to recognize a swipe gesture). Novice users tap the screen border to expose the hidden toolbar, and then swipe through the item's border from just inside the screen to outside, Figure 4.6. (We use an inside-to-outside swipe to avoid conflict with the outside-in direction used for Windows and Android system swipes, but other platforms could use either direction).

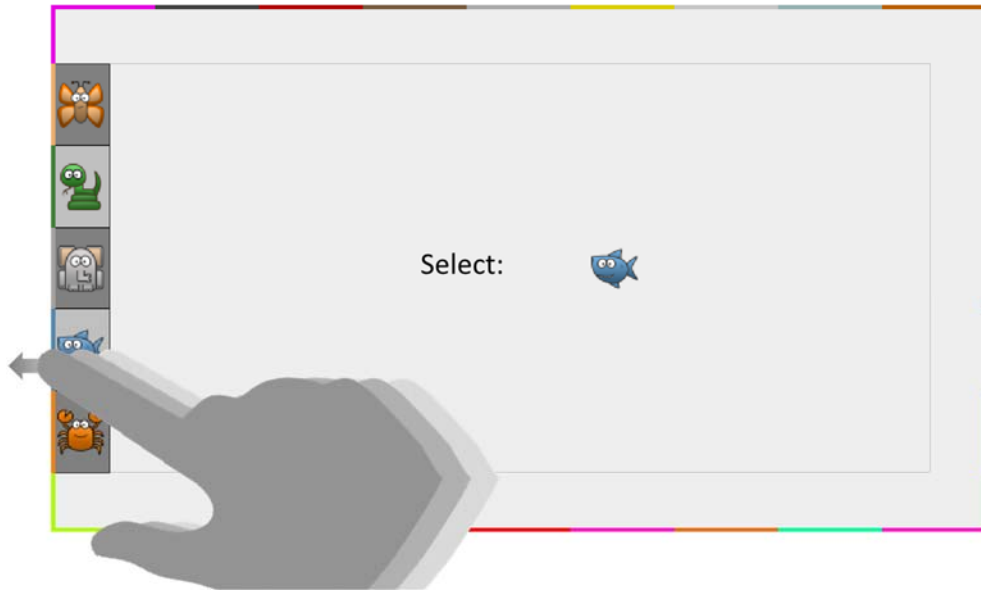


Figure 4.6. Swipe interface with left toolbar shown. Border colours indicate items boundaries. For standard selection, the user taps edge to open, then swipes item.

Once users know the item locations, they can select using only the swipe gesture (Figure 4.7). Note that taps and drag gestures in the border region are still possible – Swipe only recognizes straight inside-to-outside gestures as selections.

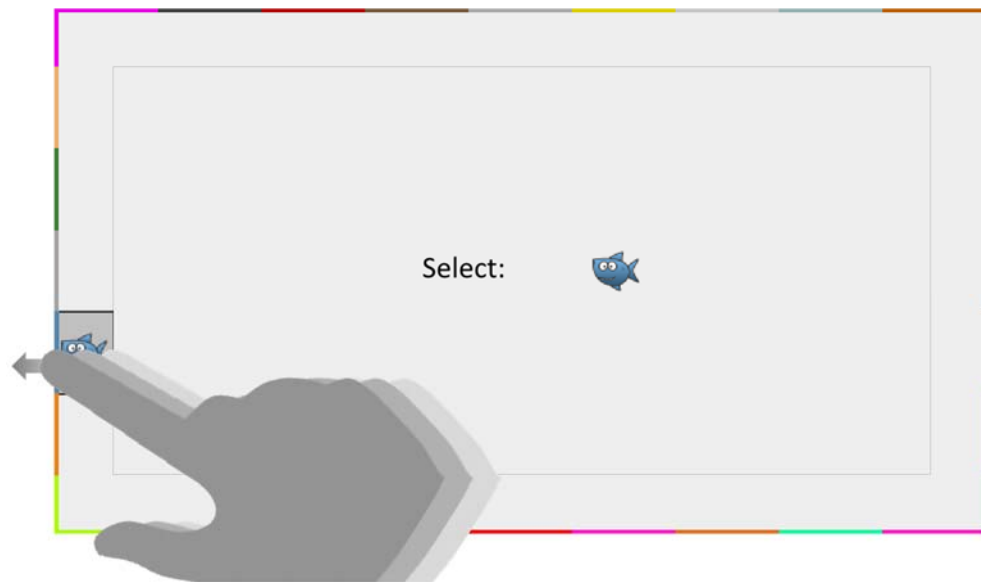


Figure 4.7. Swipe interface, shortcut selection: user swipes across item border from inside to outside.

CHAPTER 5

5 STUDY: COMPARING 4 HIDDEN-TOOLBAR DESIGNS

We compared the four designs described above in an empirical study that analyzed performance (in time and errors), learning rates for the memory-based techniques, the effect of item position on performance, and user preference. Our goal in the study is to consider initial performance after learning, rather than long-term retention (longer-term use of the interfaces is discussed later).

5.1 METHODS

5.1.1 Experimental Conditions

The four selection techniques described above (ButtonOpen, Timeout, DoubleTap, and Swipe) were built in a custom study system, implemented in Java for a Microsoft Surface 2 Pro. The main user action components of the four designs are summarized in Table 5.1 below.

Table 5.1. Differences in user actions for the four techniques.

Technique	Toolbar invocation	Item selection
ButtonOpen	Tap toolbar button	Tap item
Timeout	Tap and hold toolbar border	Tap item
DoubleTap (standard)	Tap toolbar border	Double-tap item border
DoubleTap (accelerated)	Double-tap toolbar border at item location	
Swipe (standard)	Tap toolbar border	Swipe through item border
Swipe (accelerated)	Swipe through the border at item location	

5.1.2 Procedure

Participants completed a demographics questionnaire and were introduced to the study system and the four interface techniques. Participants then sat at a table with the tablet in front of them and performed a sequence of selections with each technique. For each trial, an icon was shown in the centre of the screen, and the participant selected the command using the interface. Participants were allowed to use either hand for the selection (but they were asked to use the same hand for each trial) and were freely allowed to use either the novice or expert method; in addition, they were not required to bring their hand back to the centre of the screen between trials (but because targets were distributed around the screen, all participants did approximately centre their hand between trials). Trials were timed from the appearance of the stimulus until the target item was successfully selected. Participants were instructed to complete tasks as quickly and accurately as possible, and that any errors could be corrected simply by selecting the correct item. Completion times include any error-correction time.

Of the 28 icons in each interface, 10 were used as stimuli. Icons were chosen from controlled locations within the toolbars (corner, next-to-corner, and interior). Item locations did not change, but each method had separate sets of icons and their stimuli were in unique locations. For each interface, participants completed twelve blocks of ten selections: one practice block (data discarded) and eleven trial blocks. Interface order was fully counterbalanced and targets were presented in random order (sampling without replacement) for each block. Short rest breaks were given between blocks. After each interface, participants filled out a questionnaire based on the NASA-TLX [9]; at the end of the study, they answered questions about their preferences.

5.1.3 Participants and Apparatus

Twenty-four participants were recruited from a local university (9 female; mean age 26.3 years). All of the participants were familiar with touchscreen devices (>30 min/day). The study was conducted on a Microsoft Surface 2 Pro (10.6-inch screen, 1920×1080 resolution). Study software was written in Java and recorded all experimental data including selection times and errors.

5.1.4 Design and Research Questions

The study used a 4×11 within-participants RM-ANOVA with factors Technique (ButtonOpen, Timeout, DoubleTap, Swipe) and Block (1-11). Dependent measures were command selection time and errors per command selection. Interface order was counterbalanced between participants.

Our research questions were:

Q1. Will shortcut selection techniques for hidden toolbars be faster than the standard techniques?

Q2. Do people learn the item locations, and do they make a transition to the (memory-based) accelerated versions of the DoubleTap and Swipe techniques?

Q3. As people make the transition to the (memory-based) accelerated versions of DoubleTap and Swipe, will error rates increase?

Q4. How does item location affect selection and learning?

Q5. Which technique will be subjectively preferred?

5.2 RESULTS

5.2.1 Data Analysis

Our data analysis was based around data trials. We defined a trial as everything that occurs between the system presenting a prompt to the user and the successful selection of the menu item matching the prompt. This meant that trials include not only the successful selection but any incorrect selection (which we called errors), the time required to open a menu and perform a visual search (where relevant), and time a user spent thinking or recalling item locations.

In our analysis, we examined the means of: selection time (total time per trial), number of errors per trial, distance between menu invocation and item location, and the number of times a menu was opened per trial. No outliers (data points beyond two standard deviations of the mean) were

found for any of these measures. RM-ANOVA tests were performed to determine overall significant effects and pairwise Tukey HSD analysis for significance between selection methods.

5.2.2 Overall Selection Time

RM-ANOVA showed a significant main effect of Technique on selection time ($F_{3,69}=128.1$, $p<.001$). As seen in Figure 5.1, the Timeout technique was the slowest overall (mean 3.3 seconds per selection), with the ButtonOpen technique second (2.2 sec/selection), and the DoubleTap (1.93 sec.) and Swipe (1.98 sec.) techniques fastest.

Follow-up pairwise analysis (Tukey HSD) showed that the Timeout technique was significantly slower than all of the other techniques and that the DoubleTap and Swipe techniques were faster than ButtonOpen (all $p<.01$).

In the final block, selection times were 1517ms (Swipe), 1584ms (DoubleTap), 2233ms (ButtonOpen), and 3281ms (Timeout). The accelerated techniques were ~700ms faster than ButtonOpen and 1700ms faster than Timeout.

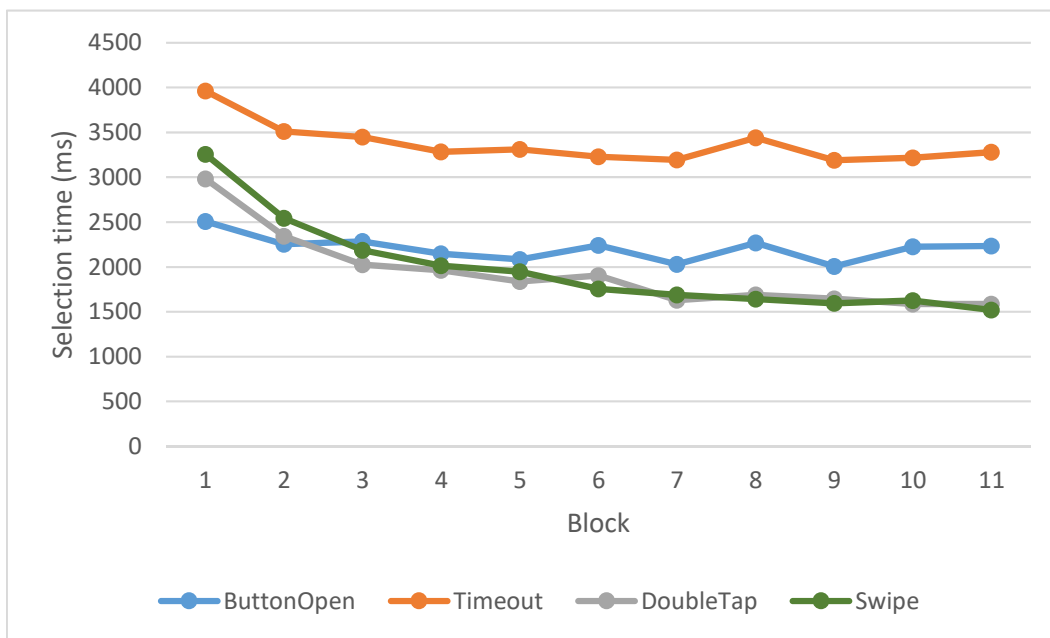


Figure 5.1. Mean selection time, by technique and block.

5.2.3 Incorrect Selection Rate

If participants selected the wrong item during a trial, they continued selecting until they chose the correct item. Overall, the mean rates of incorrect selection (see Figure 5.2) were: Swipe 5.8%; DoubleTap 6.7% ButtonOpen 0.87%; Timeout 0.76%. RM-ANOVA showed that there was a main effect of Technique on error rate ($F_{3,69}=22.1, p<.001$). There was no effect of Block ($F_{10,230}=.90, p=.54$) and no interaction ($F_{30,690}=1.23, p=.19$).

Follow-up pairwise analysis (Tukey HSD) showed that the two visually-guided techniques (ButtonOpen and Timeout) had significantly lower error rates than the two techniques with memory-based accelerators (DoubleTap and Swipe) (all $p<.01$). No other differences were found.

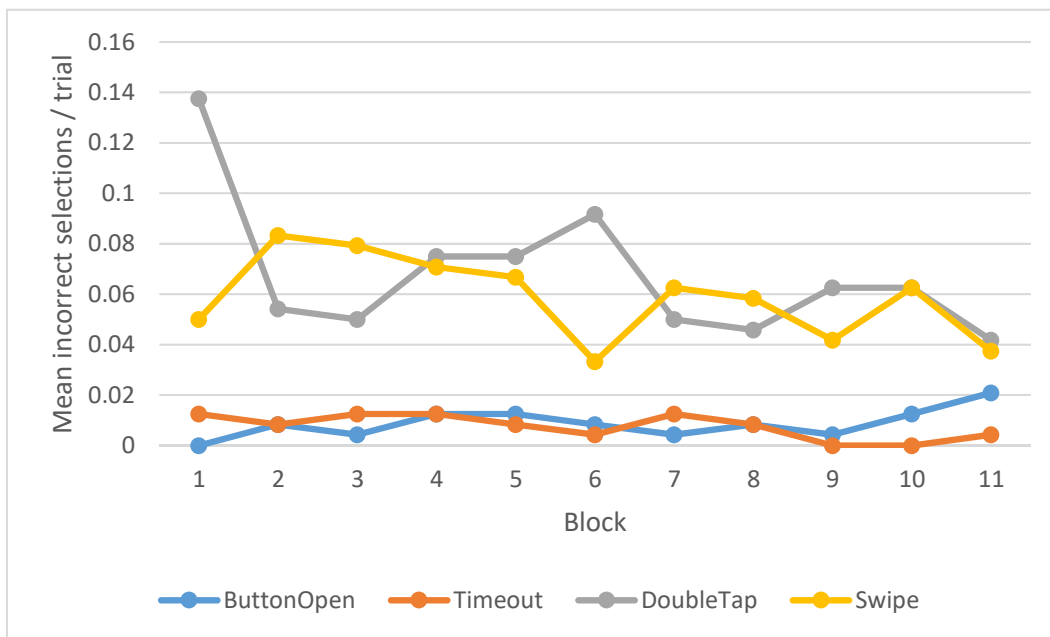


Figure 5.2. Mean incorrect selections per trial, by technique and block.

In the final block, error rates were 3.75% (Swipe), 4.17% (DoubleTap), 2.08% (ButtonOpen – higher than its overall mean of 0.87%), and 0.42% (Timeout).

5.2.4 Learning Item Locations

We recorded the location of each menu-invocation gesture (for the three techniques where the menu could be opened from anywhere on the screen edge) and calculated the distance from the invocation to the target. This provides a measure of how people learned the locations of the items over the course of the study. Results are shown in Figure 5.3.

RM-ANOVA showed significant main effects of both Block ($F_{11,253}=88.46$, $p<.001$) and Technique ($F_{2,46}=46.9$, $p<.001$). There was also an interaction ($F_{22,506}=1.89$, $p<.01$) – as shown by Figure 5.3, although people opened the menu closer and closer to the target with all three techniques, the improvement was largest with Swipe.

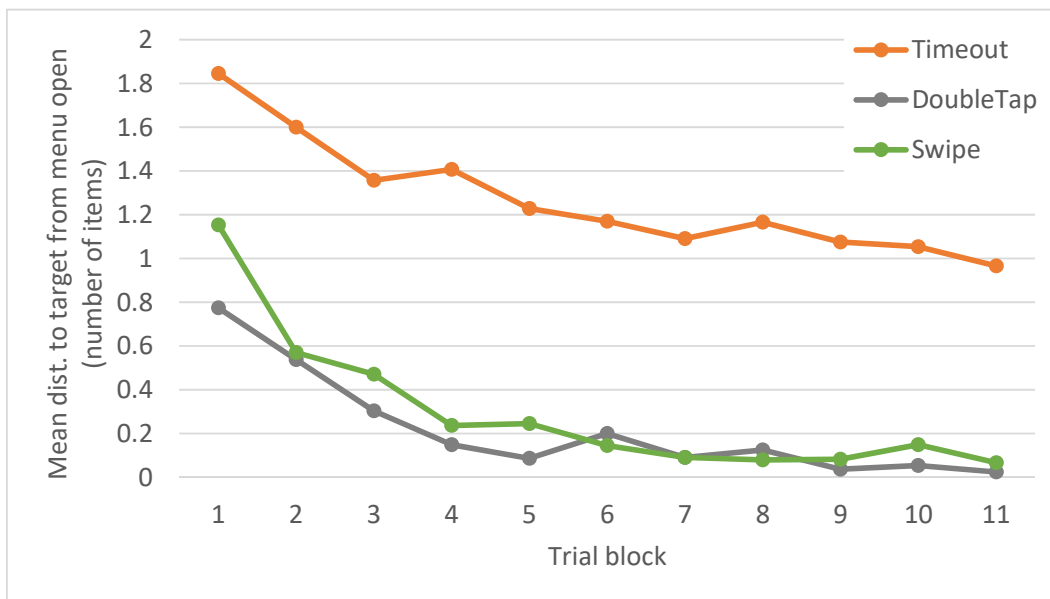


Figure 5.3. Distance from menu-invocation gesture to target item (in the number of items), by block.

5.2.5 Transition to Accelerated Method

Two of the techniques (DoubleTap and Swipe) have accelerated methods, and Figure 5.4 shows the rate at which people used the accelerated version of the technique. The rates in the final block of the study were 92% for Swipe and 88% for DoubleTap.

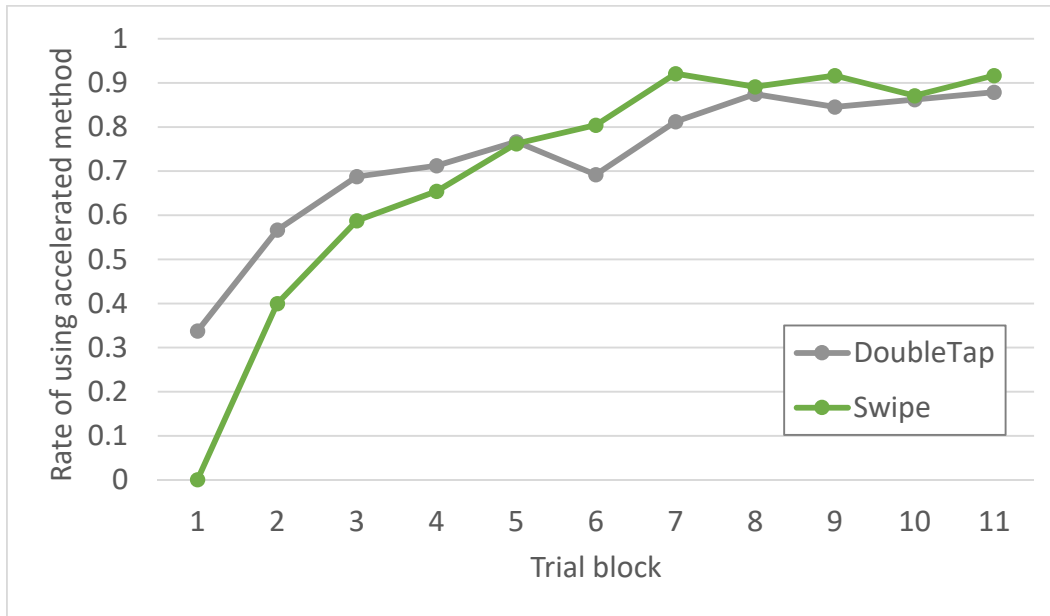


Figure 5.4. Percentage of selections made with the accelerated version of the technique, by technique and block.

RM-ANOVA showed a significant main effect of Block ($F_{11,253}=124.1$, $p<.0001$), but not of Technique ($F_{1,23}=.99$, $p=.33$). There was an interaction between the factors ($F_{11,253}=5.06$, $p<.0001$), possibly caused by the cross-over effect with Swipe’s accelerated use initially being lower than DoubleTap, but higher than DoubleTap in the later blocks.

We also compared the performance of novice and expert modes. Although both methods improved over the course of the study, the expert mode was substantially faster both in the early and the later blocks. For Swipe, novice mode improved from 3863ms per selection in the first test block to 3211ms in the final block, and expert mode improved from 2026ms to 1443ms. For DoubleTap, novice mode improved from 3982ms to 2638ms, and expert from 1968ms to 1453ms.

5.2.6 Effect of Item Locations on Learning and Performance

We analyzed the selection time and error distance by the location of the item, to determine whether different locations were faster or easier to learn. We examined specific locations, and also location classes (items in corners, items beside corners, and middle-of-toolbar items). We used only the Swipe data for this analysis since the different conditions all used different locations.

5.2.6.1 Effects on Selection Time

RM-ANOVA showed significant main effects of Location on selection time for both specific locations ($F_{9,207}=2.12, p<.05$) and location categories ($F_{2,46}=3.30, p<.05$). As shown in Figure 5.5, corner items were the fastest (mean 1861ms), and middle items were slowest (mean 2049ms). Follow-up Tukey HSD tests showed that Corner items were faster than Middle items (all $p<.05$).

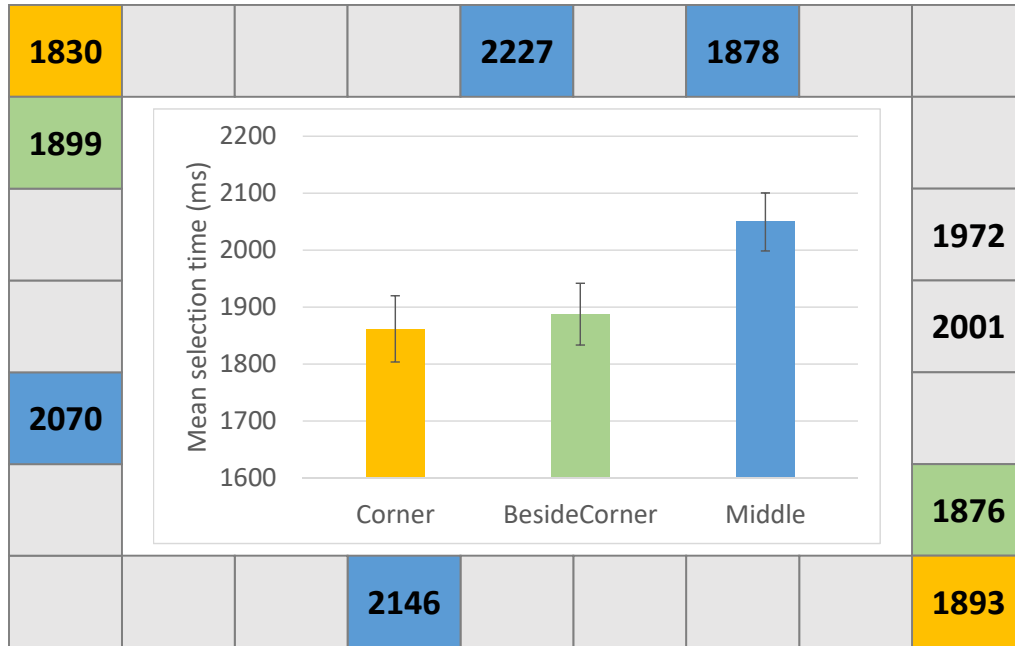


Figure 5.5. Mean selection time (Swipe data) by item location (border: actual item location; centre: location categories).

5.2.6.2 Effects on Location

RM-ANOVA showed significant effects of Location on the distance between menu invocation and target, both for specific locations ($F_{9,207}=2.82, p<.005$) and location categories ($F_{2,46}=5.20, p<.01$). Figure 5.6 shows that items next to the corners had the smallest error distance (0.23 item widths), and middle items the largest (0.33 item widths). In addition, participants used the accelerated mode more frequently for the corner and next-to-corner items (72% of selections accelerated) than for middle items (62%).

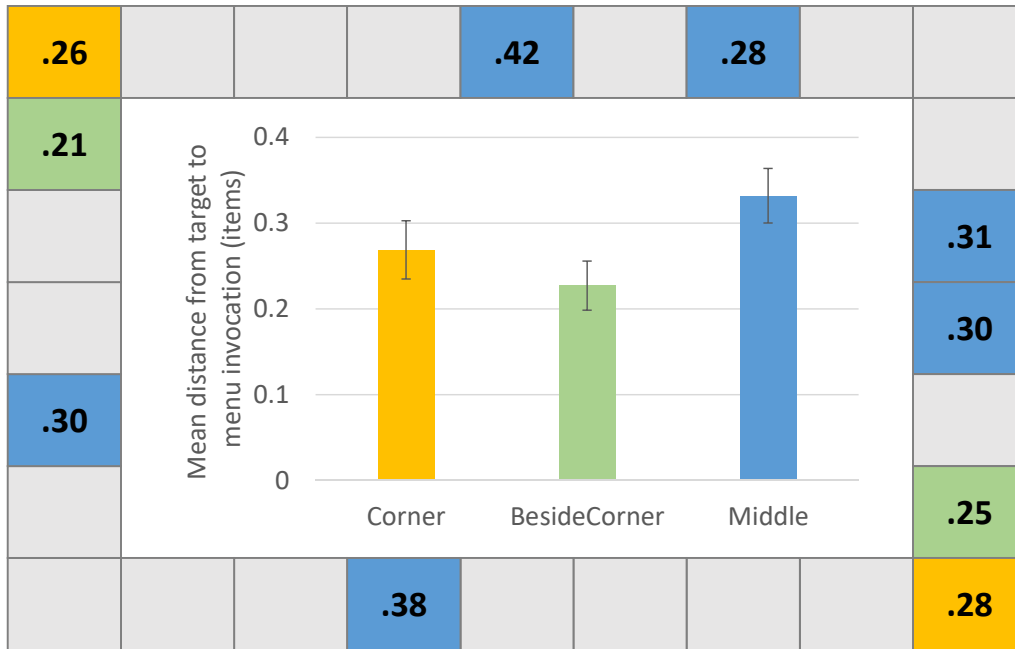


Figure 5.6. Mean distance from menu invocation to target (Swipe data), in items, by item location (border: actual item location; centre: location categories).

5.2.7 Subjective Results and Comments

After each condition, participants filled out an effort questionnaire (1-5-scale, Likert-style questions). The questions asked about the mental and physical demand required by the technique, the potential for making errors, how successful the participants felt they were with the technique and the level of frustration they felt. Results are shown in Table 5.2. Friedman tests showed only one significant difference, in responses for the question about error potential. For this question, the memory-based techniques were rated as more likely to cause errors than the visually-guided techniques.

Table 5.2. Mean (s.d.) effort questionnaire scores (1-5 scale, low to high), Friedman χ^2 value, and p-value.

	Button Open	Timeout	Double Tap	Swipe	χ_r^2	p
Mental demand	2.2 (1.1)	2.6 (1.4)	2.4 (1.0)	2.8 (1.1)	5.69	.13
Physical demand	2.2 (1.2)	3.0 (1.3)	2.2 (.98)	2.1 (1.0)	7.54	.056
Error potential	1.9 (1.1)	1.8 (1.1)	2.5 (1.1)	2.7 (1.1)	11.9	.008
Success	4.3 (.82)	4.4 (.71)	4.4 (.77)	4.2 (.64)	1.65	.65
Frustration	1.9 (.97)	2.6 (1.5)	1.8 (.92)	2.0 (1.0)	3.94	.27

At the end of the session, we also asked participants to state their preferences: we asked them to choose the technique they felt was the fastest, the most accurate, and the easiest to memorize. For the two techniques with an accelerated method, we asked with which they were more likely to use the expert mode. Finally, we asked their overall preference.

As reported in Table 5.3, Chi-squared tests showed that participants strongly preferred the Swipe technique for speed, memorability, and expert mode; for accuracy, there was no significant difference. Overall preference was also strongly in favour of the Swipe technique, with 18 of the 24 participants choosing Swipe over the other methods.

Table 5.3. - Counts of participant preferences.

	Button Open	Timeout	Double Tap	Swipe	χ^2	p
Speed	4	0	4	16	24.0	.0001
Accuracy	10	5	3	6	4.3	.23
Memorization	4	2	6	12	9.3	.025
Likely to use expert mode			5	19	8.1	.0043
Overall preference	3	0	3	18	33.0	.0001

Participant comments provide some explanations for the results presented above. For example, one person described the Swipe technique as *“good for very quick access to repeatedly used menu items. However, encourages memorization and thus prone to mistakes.”* Some people mentioned that the DoubleTap mechanism could be prone to error: *“sometimes taps would be registered as opening the menu / clicking the icon when the other was intended.”* Another participant also suggested a design improvement for DoubleTap: *“I would rather see double tap to open the menu and single tap to select because I knew where they were after only a short time.”* For the ButtonOpen technique, several comments noted problems: e.g., that the menu-invocation buttons were small (*“It was annoying to miss the small menu opening buttons”*); people also reported problems when trying to be fast with the Timeout technique (e.g., *“there was no indication of when I had held long enough for it to open, and releasing too early meant starting again.”*).

Participants also made several comments that described their ability to learn the locations. Three participants noted that the corners and nearby items were easier to learn (e.g., *“corners were the easiest to remember and some that were close to the buttons or like 2 over from a corner”*); another person stated that they used other landmarks on the device: *“on the long edges I used the webcam and Windows button as reference points for where the items where.”*). Finally, some comments suggested that people used their knowledge of item locations even for ButtonOpen: for example,

one participant stated: “*eventually I began moving my finger to the item more directly, or using another finger entirely.*”

CHAPTER 6

6 DISCUSSION

The study provides several main results:

- The two shortcut techniques (Swipe and DoubleTap) were significantly faster than the standard techniques (700ms faster than ButtonOpen, and 1700ms faster than Timeout).
- Participants quickly learned the shortcut methods and made a smooth and consistent transition to using the shortcuts for 90% of selections.
- Errors were significantly higher for both of the shortcut techniques (~5% vs. less than 1% for the visually-guided methods).
- The only difference in subjective ratings was that participants felt that the two memory-based techniques had a higher potential for errors.
- Participants strongly preferred Swipe over both of the visually-guided techniques, and also over the other shortcut technique (DoubleTap). 18 of 24 people chose Swipe overall (compared with 3 for ButtonOpen, 3 for DoubleTap, and 0 for Timeout).

In the following sections, we return to our initial research questions and discuss explanations for the results, generalization of the findings, lessons for designers of touchscreen techniques, and possibilities for future work.

6.1 ANALYSIS OF RESEARCH QUESTIONS

6.1.1 Q1. Are shortcut selection techniques faster?

The study clearly showed that both of the accelerated techniques (DoubleTap and Swipe) were faster than the methods without shortcuts. The reason for the difference seems clear – that people switched to using the shortcuts, which allowed them to make selections with fewer actions and less movement by utilizing their spatial memory. The size of the difference (700ms faster than ButtonOpen) could be meaningful in many situations – many actions on mobile devices need to be done quickly (e.g., cancelling an interruption during a meeting, or switching to the video camera from the still camera before the action finishes).

It is worth noting that people were able to improve their performance with the two standard techniques as well, but there are bottlenecks in these techniques that prevent users from achieving fast performance. For the timeout technique, even though participants learned item locations and opened the menu closer to the item when they were experienced, the delay period was clearly the limiting factor. It is possible that the timeout can be reduced to improve this technique; in addition, a visual indicator of the timeout (e.g., as used in MS Windows’ long press action) will assist users in accelerating the technique. One participant also created an ad-hoc shortcut method for ButtonOpen (by using both hands, until asked not to), but this method may not be feasible in mobile settings.

6.1.2 Q2. Do people learn the item locations, and do they make a transition to the shortcuts?

For our purposes, we considered an item to be ‘learned’ if users were able to recall where menu items were located and if this improved their performance over time. It was clear that people learned the locations since overall performance improved and even when they were still opening the menus (i.e., not using shortcuts), the distance from menu invocation dropped consistently (Figure 5.3). In addition, the rate of shortcut use climbed steadily through the study, reaching 90% by the seventh block of trials (for Swipe).

Other than the occasional misinterpretation of double taps as single taps when using DoubleTap (mentioned earlier), participants appeared to have little difficulty with the shortcut techniques. Part of the reason for their success is that both Swipe and DoubleTap provided a smooth transition from the ‘novice’ method to the shortcut. In particular, participants rehearse [21] the shortcut (e.g., swiping the bezel in the correct location) every time they execute a selection as a novice, and so switching to the shortcut method does not involve learning a new and different action (e.g., unlike other shortcut methods such as keyboard hotkeys).

6.1.3 Q3. Are error rates higher with memory-based shortcuts?

Incorrect selections were much more frequent with the shortcut techniques than with the visually-guided methods (by about 4%). This error rate is likely higher in our study than it would be in the real-world since we asked people to perform quickly, and because there were no costs to making errors (other than having to correct the selection).

The primary user problem arising from errors is in remediating the unwanted effect (e.g., replacing a mistakenly deleted object). We note that all memory-based techniques (such as keyboard shortcuts, voice commands, or gestures) are susceptible to this problem and that they remain successful regardless. To reduce the real-world costs of making errors, however, shortcut techniques such as Swipe can be used primarily for non-destructive actions (e.g., choosing tools, colours, or typestyles) that are easy to change. The technique could also be augmented to allow easy undo – for example, reversing the direction of a swipe gesture could be used to undo a previous command.

6.1.4 Q4. How does location affect learning and performance?

There were small but significant differences in the time and error distance based on location. Items in corners or next to the corners were faster (by about 200ms) than those in the middle of the toolbar. Similarly, toolbars were opened closer to corner and beside-corner locations (by about 0.8 of an item width) than middle items.

This suggests that landmarks are important for the development of spatial memory (i.e., corners are one of the strongest landmarks on a rectangular screen). We had already provided landmarks in the screen border, by colour-coding the border to the icon. Adding other landmarks may be valuable as well, as suggested by the one participant who used the webcam and windows button on the tablet bezel to help remember item locations. The work done with artificial landmarks as discussed by Uddin and colleagues could be valuable in these designs [40].

6.1.5 Q5. Which technique will be subjectively preferred?

There were strong preferences for the Swipe technique, both over the two non-shortcut techniques, and over the other accelerated technique (DoubleTap). The main reasons for people's preferences seem to be the overall performance of the technique, the ease with which people were able to make transitions to the shortcut method, and the absence of any difficulties with the shortcut mechanism (which likely reduced preference for DoubleTap).

The subjective results and participant comments show that people have strong opinions about the performance and accuracy of selection techniques – that minor problems can be painful when attempting to go fast with the interface, and that the ability to make use of experience (i.e., through shortcuts) is appreciated.

6.2 GENERALIZING AND FUTURE WORK

Here we consider how the findings can be applied to real-world situations, and how they can influence the design of selection techniques on touch interfaces. First, there are several settings in which the Swipe and DoubleTap techniques could be used with real systems. For example, swipes would be possible on the left and right sides of current Android and iOS devices, and on the top and bottom of Windows 10 tablets. Since Android already provides an application-level widget that uses a swipe gesture on the left edge (to open a menu), part of our future work is to extend this drawer widget to allow swipe-based selections. Further work is needed to determine how bezel swipes and taps will interact with system-level bezel interactions, and with application-based

gestures that operate near the bezel. For instance, an investigation like Mackay's could be performed, i.e. examine multiple interaction techniques in a real-world application by creating a set of tasks that use the same sets of actions but have different contexts, such as copying a pre-existing diagram vs. creating one from scratch [28].

Second, it is important to examine disambiguation in a system already using swiping gestures. There are existing touch screen applications that use non-bezel swiping gestures to pan across the screen (e.g., Google Maps). If our Swipe method was utilized for a menu in such an application, the system may have difficulty in distinguishing a panning swipe from a menu selection swipe. There may also be ways to mitigate this; for example, our system used swipes going from the inside of the screen out due to technical limitations but a system already using swiping gestures in the main application may be better able to discern those from a bezel swipe that starts outside the screen and moves in. It's also possible that the system could reserve any swipe gestures occurring near the edge of the screen for a menu selection swipe instead of a panning swipe. Perhaps while our studies found Swipe to be the overall preferred method, DoubleTap may prove a better method in a real-world application, especially if swiping gestures are already present. All of these factors can be examined further.

Third, the number of items that can be accommodated is an important issue. Standard hidden toolbars, because they rely on visual guidance, can have multiple rows of commands; this is not possible for a shortcut technique like Swipe, which requires that every command have a specific position on the screen edge. We plan to study how learnability changes as we add more commands to the technique. For example, we anticipate that Swipe would remain usable with twice the number of commands used in our experimental version – even with this increase, individual items would be wider than typical fingers. However, we also note that many mobile applications have small command sets and that designers often provide hidden toolbar access for a small subset of important commands, as in the Android app in Figure 1.1. If system-level interactions already use parts of the bezel (such as the top and bottom bezels for Android), there is less room for commands – but even if it is not possible to include a full set of tools, our techniques can work with a hotlist-style menu of common commands. How well the knowledge of item locations is retained over time should also be explored.

Fourth, as described above, transitions to using the shortcuts are likely to be slower in real use than observed in our experiment. However, a more gradual transition (e.g., starting with only the commands that are used most frequently) still provides an advantage over current methods – and the advantage of the rehearsal approach is that the most-frequent commands will also be learned most quickly.

Fifth, we have not yet considered how bezel interactions work when people hold a device in different positions and postures, nor have we looked at the technique for one-handed use (such as on a smartphone). Further work will explore these issues – for example, it is likely that the memorable locations on the bezel (such as the corners) are resilient to changing device orientation [38], and we believe that bezel-based interactions can still work well when used one-handed, although the area of the screen that can be used is considerably reduced and item distance may be a greater factor since the users hand will be in a fixed location. Since our work focused on a touch-based device, it is also worth examining how the techniques can be translated to mouse-based settings (e.g., using the window border rather than the bezel), how they would perform in such a system, and what other factors need to be considered in such a system (e.g., the tracking speed of the mouse especially if it is different from what a user is accustomed to).

Sixth, our system made use of coloured bars as a hint for users to remember where items were located. Since this was only meant as a hint, we did not include item colour in our analysis. However, the effect of colour on how easily a menu item is learned could be examined, as well as how many colours can be supported when it is a factor of memory. Furthermore, the visual complexity of the icons used to represent menu items and how this affects learning could also be evaluated.

Finally, it is possible that the techniques can be extended to do more than just selection. Some commands require enhanced interaction (e.g., setting a parameter in addition to selecting), and there are several possibilities for bezel interactions. For example, the length of an outside-to-inside swipe, or a drag action after a double-tap, could set a value. Similarly, “bezel chords” (multi-touch interactions in which two fingers cross the bezel at once in different locations) could substantially increase the bandwidth of the technique.

6.3 CONCLUSIONS

To examine the problem of faster selection from hidden toolbars, we identified key dimensions of this type of interaction and developed four representative hidden-toolbar designs that demonstrate different approaches. We tested these designs in an empirical study and showed that toolbars with spatial-memory-based shortcuts were significantly faster (although at the cost of more incorrect selections). Our work clearly shows that shortcuts for selections on touch interfaces can work well. There are few shortcuts available for touch interaction, and techniques such as Swipe – or shortcuts more generally – can potentially be used to improve both performance and user satisfaction. We also show that the bezel is an under-exploited resource on many touch devices that could be used to provide higher-bandwidth interaction for experts, and that spatial memory is a powerful tool for shortcuts, particularly when combined with rehearsals to enable smooth transitions to expertise.

REFERENCES

1. Accot, J. and Zhai, S. Beyond Fitts' Law: Models for Trajectory-Based HCI Tasks. In *Proceedings of CHI'97 Conference on Human Factors in Computing Systems* (1997), 295-302.
2. Accot, J. and Zhai, S. More than Dotting the i's -- Foundations for Crossing-Based Interfaces. In *Proceedings of CHI'2002 Conference on Human Factors in Computing Systems, Minneapolis, Minnesota, 20--25 April 2002*, 73-80.
3. Anderson, F. and Bischof, W.F. Learning and performance with gesture guides. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*, ACM, (2013), 1109-1118.
4. Apitz, G. and Guimbretiere, F. CrossY: a crossing-based drawing application. In *Proceedings of the 17th annual ACM symposium on User interface software and technology*, ACM, Santa Fe, NM, USA, 2004, 3-12.
5. Au, Irene, and Shuang Li. Netscape Communicator's Collapsible Toolbars. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 1998, 81-86.
6. Bailly, G., Lecolinet, E. and Guiard, Y. Finger-Count and Radial-Stroke: Two Techniques for Augmenting Linear Menus on Multi-Touch Surfaces. In *Proceedings of CHI 2010: ACM Conference on Human Factors in Computing Systems*, ACM Press, (2010), 591-594.
7. Bau, O. and Mackay, W.E. OctoPocus: a dynamic guide for learning gesture-based command sets. In *Proceedings of the ACM Symposium on User Interface Software and Technology*, ACM, (2008), 37-46.
8. Bragdon, A., Nelson, E., Li, Y. and Hinckley, K. Experimental analysis of touch-screen gesture designs in mobile environments. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM, Vancouver, BC, Canada, 2011, 403-412.
9. Cockburn, A. and Gutwin, C. A Predictive Model of Human Performance with Scrolling and Hierarchical Lists. In *Human-Computer Interaction* 24, 3 (2009), 273-314.
10. Cockburn, A., Gutwin, C., Scarr, J. and Malacria, S. Supporting Novice to Expert Transitions in User Interfaces. In *ACM Computing Surveys* 47, 2 (2014), 31:31-31:36.
11. Cockburn, A., Kristensson, P., Alexander, J. and Zhai, S. Hard Lessons: Effort-Inducing Interfaces Benefit Spatial Learning. In *Proceedings of CHI'07: ACM Conference on Human Factors in Computing Systems*, ACM Press, (2007), 1571-1580.
12. Ehret, B. Learning Where to Look: Location Learning in Graphical User Interfaces. In *Proceedings of CHI'2002 Conference on Human Factors in Computing Systems*, ACM Press, (2002), 211-218.

13. Fitts, P. and Melton, A. The Information Capacity of the Human Motor System in Controlling the Amplitude of Movement. In *Journal of Experimental Psychology* 47, no. 6 (1954): 381-91.
14. Froehlich, J., Wobbrock, J.O. and Kane, S.K. Barrier pointing: using physical edges to assist target acquisition on mobile device touch screens. In *Proceedings of the 9th international ACM SIGACCESS conference on Computers and accessibility*, ACM, Tempe, Arizona, USA, 2007, 19-26.
15. Ghomi, E., Faure, G., St, Huot, p., Chapuis, O. and Beaudouin-Lafon, M. Using rhythmic patterns as an input method. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM, Austin, Texas, USA, 2012, 1253-1262.
16. Goldberg, D. and Richardson, C. Touch-Typing With a Stylus. In *Proceedings of INTERCHI'93 Conference on Human Factors in Computing Systems*, (1993), 80-87.
17. Gutwin, C., Cockburn, A., Scarr, J. and Malacria, S. Faster Command Selection on Tablets with FastTap. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM, (2014), 2617-2626.
18. Hinckley, K., Pierce, J., Sinclair, M. and Horvitz, E. Sensing techniques for mobile interaction. In *Proceedings of the 2000 ACM Symposium on User Interface Software and Technology*, (2000), 91-100.
19. 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*, ACM, New York, New York, USA, 2010, 27-36.
20. Jain, M. and Balakrishnan, R. User learning and performance with bezel menus. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM, Austin, Texas, USA, 2012, 2221-2230.
21. Kurtenbach, G. The Design and Evaluation of Marking Menus. In *Department of Computer Science*, University of Toronto, Toronto, 1993.
22. Kurtenbach, G. and Buxton, B. The Limits of Expert Performance Using Hierarchic Marking Menus. In *Proceedings of InterCHI'93*, (1993), 482-487.
23. Kurtenbach, G. and Buxton, W. User Learning and Performance with Marking Menus. In *Proceedings of CHI'94 Conference on Human Factors in Computing Systems*, ACM, (1994), 258-264.
24. Kurtenbach, G., Fitzmaurice, G., Baudel, T. and Buxton, B. The Design of a GUI Paradigm based on Tablets, Two-hands, and Transparency. In *Proceedings of CHI'97 Conference on Human Factors in Computing Systems*, Atlanta, Georgia, March 22-27, (1997), 35--42.

25. Kurtenbach, G., Fitzmaurice, G., Owen, R. and Baudel, T. The Hotbox: efficient access to a large number of menu-items. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, ACM, Pittsburgh, Pennsylvania, USA, 1999, 231-237.
26. Kurtenbach, G., Sellen, A. and Buxton, W. An Empirical Evaluation of Some Articulatory and Cognitive Aspects of Marking Menus. In *Human-Computer Interaction 8*, 1 (1993), 1-23.
27. Li, Y., Hinckley, K., Guan, Z., & Landay, J. A. Experimental analysis of mode switching techniques in pen-based user interfaces. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, ACM, Portland, Oregon, USA, 2005, 461-470.
28. Mackay, W.E. Which interaction technique works when?: floating palettes, marking menus and toolglasses support different task strategies. In *Proceedings of the Working Conference on Advanced Visual Interfaces*, ACM, Trento, Italy, 2002, 203-208.
29. Malacria, S., Bailly, G., Harrison, J., Cockburn, A. and Gutwin, C. Promoting Hotkey Use through Rehearsal with ExposeHK. In *Proceedings of CHI'13: ACM Conference on Human Factors in Computing Systems*, ACM, (2013), 573-582.
30. Ramos, G. and Balakrishnan, R. Zliding: Fluid Zooming and Sliding for High Precision Parameter Manipulation. In *Proceedings of UIST'05: ACM Symposium on User Interface Software and Technology*, ACM Press, (2005), 143-152.
31. Ramos, G., Buoulos, M. and Balakrishnan, R. Pressure Widgets. In *Proceedings of CHI'04. ACM Conference on Human Factors in Computing Systems*, ACM Press, (2004), 487-494.
32. Robertson, G., Czerwinski, M., Larson, K., Robbins, D., Thiel, D. and van Dantzich, M. Data Mountain: Using Spatial Memory for Document Management. In *Proceedings of UIST'98 ACM Symposium on User Interface Software and Technology*, (1998), 153-162.
33. Roth, V. and Turner, T. Bezel swipe: conflict-free scrolling and multiple selection on mobile touch screen devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM, Boston, MA, USA, 2009, 1523-1526.
34. Roudaut, A., Lecolinet, E. and Guiard, Y. MicroRolls: Expanding Touch-Screen Input Vocabulary by Distinguishing Rolls vs. Slides of the Thumb. In *Proceedings of CHI 2009: ACM Conference on Human Factors in Computing Systems*, ACM, (2009), 1927-1936.
35. Scarr, J., Cockburn, A. and Gutwin, C. Supporting and Exploiting Spatial Memory in User Interfaces. In *Foundations and Trends in Human-Computer Interaction 6*, 1 (2013), 1-84
36. Scarr, J., Cockburn, A., Gutwin, C. and Bunt, A. Improving command selection with CommandMaps. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM, (2012), 257-266.

37. Scarr, J., Cockburn, A., Gutwin, C., Bunt, A. and Cechanowicz, J. The Usability of CommandMaps in Realistic Tasks. In *Proceedings of ACM CHI 2014 Conference on Human Factors in Computing Systems*, ACM, (2014).
38. Scarr, J., Cockburn, A., Gutwin, C. and Malacria, S. Testing the robustness and performance of spatially consistent interfaces. In *Proceedings of the 2013 SIGCHI Conference on Human Factors in Computing Systems*, ACM, (2013), 3139-3148.
39. Schmidt, R. Frequent augmented feedback can degrade learning: Evidence and interpretations. in Requin, J. and Stelmach, G. eds. In *Tutorials in motor neuroscience*, 1991, 59-75.
40. Uddin, M., Gutwin, C., Cockburn, A. The Effects of Artificial Landmarks on Learning and Performance in Spatial-Memory Interfaces. In *Proceedings of the 2017 SIGCHI Conference on Human Factors in Computing Systems*, ACM, Denver, CO, USA, (2017), 3843-3855.
41. Wickens, C. and Carswell, C. Proximity Compatibility Principle: Its Psychological Foundation and Relevance to Display Design. In *Human Factors* 37, no. 3 (1995): 473-94.
42. Wobbrock, J., Myers, B. and Kembel, J. EdgeWrite: A stylus-based text entry method designed for high accuracy and stability of motion. In *Proceedings of the 16th annual ACM symposium on User Interface Software and Technology*, ACM Press, (2003), 61-70.
43. Yu, N.-H., Huang, D.-Y., Hsu, J.-J. and Hung, Y.-P. Rapid selection of hard-to-access targets by thumb on mobile touch-screens. In *Proceedings of the 15th international conference on Human-computer interaction with mobile devices and services*, ACM, Munich, Germany, 2013, 400-403.
44. Zhai, S. and Kristensson, P. Shorthand Writing on Stylus Keyboard. In *Proceedings of CHI'2003 Conference on Human Factors in Computing Systems*, (2003), 97-104.

APPENDIX A INITIAL DESIGNS STUDY

A.1 CONSENT FORM

DEPARTMENT OF COMPUTER SCIENCE
UNIVERSITY OF SASKATCHEWAN
INFORMED CONSENT FORM



Research Project: **Gate Crossing Interaction Technique**

Investigators: Dr. Carl Gutwin, Department of Computer Science (966-8646)

Katherine Schramm, Department of Computer Science

This consent form, a copy of which has been given to you, is only part of the process of informed consent. It should give you the basic idea of what the research is about and what your participation will involve. If you would like more detail about something mentioned here, or information not included here, please ask. Please take the time to read this form carefully and to understand any accompanying information.

This study is concerned with **measuring simulated expert behaviour use of two interaction methods with a menu system.**

The goal of the research is to **determine if using a gate-crossing interaction technique is more efficient with a fade-in menu system than a standard click is.**

The session will require **40** minutes, during which you will be asked to **perform a prompted set of menu selections using two different methods on both a tablet and a PC** in the Human-Computer Interaction Lab at the University of Saskatchewan.

At the end of the session, you will be given more information about the purpose and goals of the study, and there will be time for you to ask questions about the research. As a way of thanking you for your participation and to help compensate you for your time and any travel costs you may have incurred, you will receive a **\$10** honorarium at the end of the session.

The data collected from this study will be used in articles for publication in journals and conference proceedings.

As one way of thanking you for your time, we will be pleased to make available to you a summary of the results of this study once they have been compiled (usually within two months). This summary will outline the research and discuss our findings and recommendations. This summary will be available on the HCI lab's website: <http://www.hci.usask.ca/>

All personal and identifying data will be kept confidential. Confidentiality will be preserved by using pseudonyms in any presentation of textual data in journals or at conferences. The informed consent form and all research data will be kept in a secure location under confidentiality in accordance with University policy for 5 years post-publication. Do you have any questions about this aspect of the study?

You are free to withdraw from the study at any time without penalty and without losing any advertised benefits. Withdrawal from the study will not affect your academic status or your access to services at the university. If you withdraw, your data will be deleted from the study and destroyed. Your right to withdraw data from the study will apply until results have been disseminated, data has been pooled, etc. After this, it is possible that some form of research dissemination will have already occurred and it may not be possible to withdraw your data.

Your continued participation should be as informed as your initial consent, so you should feel free to ask for clarification or new information throughout your participation. If you have further questions concerning matters related to this research, please contact:

- Dr. Carl Gutwin, Professor, Dept. of Computer Science, (306) 966-8646, gutwin@cs.usask.ca

Your signature on this form indicates that you have understood to your satisfaction the information regarding participation in the research project and agree to participate as a participant. In no way does this waive your legal rights nor release the investigators, sponsors, or involved institutions from their legal and professional responsibilities. If you have further questions about this study or your rights as a participant, please contact:

- Dr. Carl Gutwin, Professor, Dept. of Computer Science, (306) 966-8646, gutwin@cs.usask.ca
- Research Ethics Office, University of Saskatchewan, (306) 966-2975 or toll-free at 888-966-2975.

Participant's signature: _____

Date: _____

Investigator's signature: _____

Date: _____

A copy of this consent form has been given to you to keep for your records and reference. This research has the ethical approval of the Research Ethics Office at the University of Saskatchewan.

A.2 DEMOGRAPHIC FORM

Demographics

Age					
Sex	Male			Female	
Handedness	Right			Left	
How often do you use a computer?	Never	Rarely	Occasionally	Often	Everyday
How often do you use a mobile phone?	Never	Rarely	Occasionally	Often	Everyday
How often do you use a tablet device?	Never	Rarely	Occasionally	Often	Everyday
How often do you use a touch surface?	Never	Rarely	Occasionally	Often	Everyday

A.3 TLX FORMS

Tablet Clicks

How mentally demanding was the selection method?	1 Low	2	3	4	5 High
How physically demanding was the selection method?	1 Low	2	3	4	5 High
How easy was it to make mistakes with the selection method?	1 Low	2	3	4	5 High
How successful were you in accomplishing tasks using the selection method?	1 Low	2	3	4	5 High
How insecure, discouraged, irritated, stressed, and annoyed were you with the selection method?	1 Low	2	3	4	5 High
Any comments?					

Tablet Gates

How mentally demanding was the selection method?	1 Low	2	3	4	5 High
How physically demanding was the selection method?	1 Low	2	3	4	5 High
How easy was it to make mistakes with the selection method?	1 Low	2	3	4	5 High
How successful were you in accomplishing tasks using the selection method?	1 Low	2	3	4	5 High
How insecure, discouraged, irritated, stressed, and annoyed were you with the selection method?	1 Low	2	3	4	5 High
Any comments?					

PC Clicks

How mentally demanding was the selection method?	1 Low	2	3	4	5 High
How physically demanding was the selection method?	1 Low	2	3	4	5 High
How easy was it to make mistakes with the selection method?	1 Low	2	3	4	5 High
How successful were you in accomplishing tasks using the selection method?	1 Low	2	3	4	5 High
How insecure, discouraged, irritated, stressed, and annoyed were you with the selection method?	1 Low	2	3	4	5 High
Any comments?					

PC Gates

How mentally demanding was the selection method?	1 Low	2	3	4	5 High
How physically demanding was the selection method?	1 Low	2	3	4	5 High
How easy was it to make mistakes with the selection method?	1 Low	2	3	4	5 High
How successful were you in accomplishing tasks using the selection method?	1 Low	2	3	4	5 High
How insecure, discouraged, irritated, stressed, and annoyed were you with the selection method?	1 Low	2	3	4	5 High
Any comments?					

Tablet Overall

With which interface did you perform faster?	Mostly Clicks	Slightly Clicks	Both were the same	Slightly Gates	Mostly Gates
With which interface did you make fewer mistakes?	Mostly Clicks	Slightly Clicks	Both were the same	Slightly Gates	Mostly Gates
With which did you find it easier to learn the locations of menu items?	Mostly Clicks	Slightly Clicks	Both were the same	Slightly Gates	Mostly Gates
With which did you find it easier to learn the expert behaviour?	Mostly Clicks	Slightly Clicks	Both were the same	Slightly Gates	Mostly Gates
Which interface did you prefer overall?	Mostly Clicks	Slightly Clicks	Both were the same	Slightly Gates	Mostly Gates
Any comments?					

PC Overall

With which interface did you perform faster?	Mostly Clicks	Slightly Clicks	Both were the same	Slightly Gates	Mostly Gates
With which interface did you make fewer mistakes?	Mostly Clicks	Slightly Clicks	Both were the same	Slightly Gates	Mostly Gates
With which did you find it easier to learn the locations of menu items?	Mostly Clicks	Slightly Clicks	Both were the same	Slightly Gates	Mostly Gates
With which did you find it easier to learn the expert behaviour?	Mostly Clicks	Slightly Clicks	Both were the same	Slightly Gates	Mostly Gates
Which interface did you prefer overall?	Mostly Clicks	Slightly Clicks	Both were the same	Slightly Gates	Mostly Gates
Any comments?					

APPENDIX B SECOND DESIGNS STUDY

B.1 CONSENT FORM

DEPARTMENT OF COMPUTER SCIENCE
UNIVERSITY OF SASKATCHEWAN
INFORMED CONSENT FORM



Research Project: **Gate Crossing Menus**

Investigators: Dr. Carl Gutwin, Department of Computer Science (966-8646)
Katherine Schramm, Department of Computer Science

This consent form, a copy of which has been given to you, is only part of the process of informed consent. It should give you the basic idea of what the research is about and what your participation will involve. If you would like more detail about something mentioned here, or information not included here, please ask. Please take the time to read this form carefully and to understand any accompanying information.

This study is concerned with **measuring the performances of four methods of interacting with a menu system.**

The goal of the research is to **determine if using a gate-crossing interaction technique is more efficient with a fade-in menu system than a standard click is.**

The session will require **40** minutes, during which you will be asked to **perform a series of prompted menu selections using four different selection methods on a touch tablet** in the Human-Computer Interaction Lab at the University of Saskatchewan.

At the end of the session, you will be given more information about the purpose and goals of the study, and there will be time for you to ask questions about the research.

The data collected from this study will be used in articles for publication in journals and conference proceedings.

As one way of thanking you for your time, we will be pleased to make available to you a summary of the results of this study once they have been compiled (usually within two months). This summary will outline the research and discuss our findings and recommendations. This summary will be available on the HCI lab's website: <http://www.hci.usask.ca/>

All personal and identifying data will be kept confidential. Confidentiality will be preserved by using pseudonyms in any presentation of textual data in journals or at conferences. The informed consent form and all research data will be kept in a secure location under confidentiality in accordance with University policy for 5 years post-publication. Do you have any questions about this aspect of the study?

You are free to withdraw from the study at any time without penalty and without losing any advertised benefits. Withdrawal from the study will not affect your academic status or your access to services at the university. If you withdraw, your data will be deleted from the study and destroyed. Your right to withdraw data from the study will apply until results have been disseminated, data has been pooled, etc. After this, it is possible that some form of research dissemination will have already occurred and it may not be possible to withdraw your data.

Your continued participation should be as informed as your initial consent, so you should feel free to ask for clarification or new information throughout your participation. If you have further questions concerning matters related to this research, please contact:

- Dr. Carl Gutwin, Professor, Dept. of Computer Science, (306) 966-8646, gutwin@cs.usask.ca

Your signature on this form indicates that you have understood to your satisfaction the information regarding participation in the research project and agree to participate as a participant. In no way does this waive your legal rights nor release the investigators, sponsors, or involved institutions from their legal and professional responsibilities. If you have further questions about this study or your rights as a participant, please contact:

- Dr. Carl Gutwin, Professor, Dept. of Computer Science, (306) 966-8646, gutwin@cs.usask.ca
- Research Ethics Office, University of Saskatchewan, (306) 966-2975 or toll-free at 888-966-2975.

Participant's signature: _____

Date: _____

Investigator's signature: _____

Date: _____

A copy of this consent form has been given to you to keep for your records and reference. This research has the ethical approval of the Research Ethics Office at the University of Saskatchewan.

B.2 DEMOGRAPHIC FORM

Demographics

Age					
Sex	Male		Female		
Handedness	Right		Left		
How often do you use a computer?	Never	Rarely	Occasionally	Often	Everyday
How often do you use a mobile phone?	Never	Rarely	Occasionally	Often	Everyday
How often do you use a tablet device?	Never	Rarely	Occasionally	Often	Everyday
How often do you use a touch surface?	Never	Rarely	Occasionally	Often	Everyday
How often do you play video games?	Never	Rarely	Occasionally	Often	Everyday

B.3 TLX FORMS

Click					
How mentally demanding was the selection method?	1 Low	2	3	4	5 High
How physically demanding was the selection method?	1 Low	2	3	4	5 High
How easy was it to make mistakes with the selection method?	1 Low	2	3	4	5 High
How successful were you in accomplishing tasks using the selection method?	1 Low	2	3	4	5 High
How insecure, discouraged, irritated, stressed, and annoyed were you with the selection method?	1 Low	2	3	4	5 High
Any comments?					

Hold-and-Click					
How mentally demanding was the selection method?	1 Low	2	3	4	5 High
How physically demanding was the selection method?	1 Low	2	3	4	5 High
How easy was it to make mistakes with the selection method?	1 Low	2	3	4	5 High
How successful were you in accomplishing tasks using the selection method?	1 Low	2	3	4	5 High
How insecure, discouraged, irritated, stressed, and annoyed were you with the selection method?	1 Low	2	3	4	5 High
Any comments?					

Double Click					
How mentally demanding was the selection method?	1 Low	2	3	4	5 High

How physically demanding was the selection method?	1 Low	2	3	4	5 High
How easy was it to make mistakes with the selection method?	1 Low	2	3	4	5 High
How successful were you in accomplishing tasks using the selection method?	1 Low	2	3	4	5 High
How insecure, discouraged, irritated, stressed, and annoyed were you with the selection method?	1 Low	2	3	4	5 High
Any comments?					

Gate

How mentally demanding was the selection method?	1 Low	2	3	4	5 High
How physically demanding was the selection method?	1 Low	2	3	4	5 High
How easy was it to make mistakes with the selection method?	1 Low	2	3	4	5 High
How successful were you in accomplishing tasks using the selection method?	1 Low	2	3	4	5 High
How insecure, discouraged, irritated, stressed, and annoyed were you with the selection method?	1 Low	2	3	4	5 High
Any comments?					

Overall

With which interface did you perform faster?	Click	Hold-and-Click	Double Click	Gate
With which interface did you make fewer mistakes?	Click	Hold-and-Click	Double Click	Gate
With which did you find it easier to learn the locations of menu items?	Click	Hold-and-Click	Double Click	Gate
With which did you find it easier to select items without opening a menu first?	Double Click		Gate	

Which interface did you prefer overall?	Click	Hold-and-Click	Double Click	Gate
Any comments?				

APPENDIX C FINAL DESIGNS STUDY

C.1 CONSENT FORM

DEPARTMENT OF COMPUTER SCIENCE
UNIVERSITY OF SASKATCHEWAN
INFORMED CONSENT FORM



Research Project: **Gate Crossing Menus**

Investigators: Dr. Carl Gutwin, Department of Computer Science (966-8646)

Katherine Schramm, Department of Computer Science

This consent form, a copy of which has been given to you, is only part of the process of informed consent. It should give you the basic idea of what the research is about and what your participation will involve. If you would like more detail about something mentioned here, or information not included here, please ask. Please take the time to read this form carefully and to understand any accompanying information.

This study is concerned with **measuring the performances of four methods of interacting with a menu system and transitions between novice and expert behaviour.**

The goal of the research is to **determine if a gate-crossing interaction technique is the most efficient way to interact with a fade-in menu system.**

The session will require **45** minutes, during which you will be asked to **perform a series of prompted menu selections using four different selection methods on a touch tablet** in the Human-Computer Interaction Lab at the University of Saskatchewan.

At the end of the session, you will be given more information about the purpose and goals of the study, and there will be time for you to ask questions about the research. As a way of thanking you for your participation and to help compensate you for your time and any travel costs you may have incurred, you will receive a **\$10** honorarium at the end of the session.

The data collected from this study will be used in articles for publication in journals and conference proceedings.

As one way of thanking you for your time, we will be pleased to make available to you a summary of the results of this study once they have been compiled (usually within two months). This summary will outline the research and

discuss our findings and recommendations. This summary will be available on the HCI lab's website: <http://www.hci.usask.ca/>

All personal and identifying data will be kept confidential. Confidentiality will be preserved by using pseudonyms in any presentation of textual data in journals or at conferences. The informed consent form and all research data will be kept in a secure location under confidentiality in accordance with University policy for 5 years post-publication. Do you have any questions about this aspect of the study?

You are free to withdraw from the study at any time without penalty and without losing any advertised benefits.

Withdrawal from the study will not affect your academic status or your access to services at the university. If you withdraw, your data will be deleted from the study and destroyed. Your right to withdraw data from the study will apply until results have been disseminated, data has been pooled, etc. After this, it is possible that some form of research dissemination will have already occurred and it may not be possible to withdraw your data.

Your continued participation should be as informed as your initial consent, so you should feel free to ask for clarification or new information throughout your participation. If you have further questions concerning matters related to this research, please contact:

- Dr. Carl Gutwin, Professor, Dept. of Computer Science, (306) 966-8646, gutwin@cs.usask.ca

Your signature on this form indicates that you have understood to your satisfaction the information regarding participation in the research project and agree to participate as a participant. In no way does this waive your legal rights nor release the investigators, sponsors, or involved institutions from their legal and professional responsibilities. If you have further questions about this study or your rights as a participant, please contact:

- Dr. Carl Gutwin, Professor, Dept. of Computer Science, (306) 966-8646, gutwin@cs.usask.ca
- Research Ethics Office, University of Saskatchewan, (306) 966-2975 or toll-free at 888-966-2975.

Participant's signature: _____

Date: _____

Investigator's signature: _____

Date: _____

A copy of this consent form has been given to you to keep for your records and reference. This research has the ethical approval of the Research Ethics Office at the University of Saskatchewan.

research has the ethical approval of the Research Ethics Office at the University of Saskatchewan.

research has the ethical approval of the Research Ethics Office at the University of Saskatchewan.

C.2 DEMOGRAPHIC FORM

GATE CROSSING MENUS: DEMOGRAPHIC FORM

*Required

Participant ID *

Age *

Gender *

- Male
- Female
- Other

Handedness *

- Right
- Left
- Other

On average, how much time do you spend on computers a day? *

- Less than 30 minutes
- 30-60 minutes
- 1-2 hours
- 2-4 hours
- 4-8 hours
- More than 8 hours

On average, how much time do you spend on mobile phones a day? *

- Less than 30 minutes
- 30-60 minutes
- 1-2 hours
- 2-4 hours
- 4-8 hours
- More than 8 hours

How much time do you spend using a tablet device? *

- None
- Less than 3 hours a week
- 3-7 hours a week
- 1-2 hours a day
- More than 2 hours a day

How much time do you spend using a touch surface? *






e.g. mobile phone with touch screen, touch screen laptop, tablet etc.

- None
- Less than 3 hours a week
- 3-7 hours a week
- 1-2 hours a day
- More than 2 hours a day

Please list the touch surface devices you use

Please list some of the touch device apps that you most frequently use

How much time do you spend playing video games? *

-  None
-  Less than 3 hours a week
-  3-7 hours a week
-  1-2 hours a day
-  More than 2 hours a day

Please list some of the video games you most frequently play

C.3 TLX FORMS

GATE CROSSING MENUS: CLICK QUESTIONNAIRE

The following questions pertain to the click method.

***Required**

Participant ID *

How mentally demanding was the selection method? *

1 2 3 4 5

Low High

How physically demanding was the selection method? *

1 2 3 4 5

Low High

How easy was it to make mistakes with the selection method? *

1 2 3 4 5

Low High

How successful were you in accomplishing tasks using the selection method? *

1 2 3 4 5

Low High

How insecure, discouraged, irritated, stressed, and annoyed were you with the selection method? *

1 2 3 4 5

Low High

Any Comments?



GATE CROSSING MENUS: HOLD-AND-CLICK QUESTIONNAIRE

The following questions pertain to the hold-and-click method.

***Required**

Participant ID *

How mentally demanding was the selection method? *

1 2 3 4 5

Low High

How physically demanding was the selection method? *

1 2 3 4 5

Low High

How easy was it to make mistakes with the selection method? *

1 2 3 4 5

Low High

How successful were you in accomplishing tasks using the selection method? *

1 2 3 4 5

Low High

How insecure, discouraged, irritated, stressed, and annoyed were you with the selection method? *

1 2 3 4 5

Low High

Any Comments?



GATE CROSSING MENUS: DOUBLE CLICK QUESTIONNAIRE

The following questions pertain to the double click method.

***Required**

Participant ID *

How mentally demanding was the selection method? *

1 2 3 4 5

Low High

How physically demanding was the selection method? *

1 2 3 4 5

Low High

How easy was it to make mistakes with the selection method? *

1 2 3 4 5

Low High

How successful were you in accomplishing tasks using the selection method? *

1 2 3 4 5

Low High

How insecure, discouraged, irritated, stressed, and annoyed were you with the selection method? *

1 2 3 4 5

Low High

Any Comments?



GATE CROSSING MENUS: GATE QUESTIONNAIRE

The following questions pertain to the gate method.

***Required**

Participant ID *

How mentally demanding was the selection method? *

1 2 3 4 5

Low High

How physically demanding was the selection method? *

1 2 3 4 5

Low High

How easy was it to make mistakes with the selection method? *

1 2 3 4 5

Low High

How successful were you in accomplishing tasks using the selection method? *

1 2 3 4 5

Low High

How insecure, discouraged, irritated, stressed, and annoyed were you with the selection method? *

1 2 3 4 5

Low High

Any Comments?



GATE CROSSING MENUS: POST-QUESTIONNAIRE

*Required

Participant ID *

With which interface did you perform faster? *

- Click
- Hold-and-Click
- Double Click
- Gate

With which interface did you make fewer mistakes? *

- Click
- Hold-and-Click
- Double Click
- Gate

With which did you find it easier to learn the locations of menu items? *

- Click
- Hold-and-Click
- Double Click
- Gate

With which did you find it easier to select items without opening a menu first? *

- Double Click
- Gate

Which interface did you prefer overall? *

- Click
- Hold-and-Click
- Double Click
- Gate

Did you notice that the colours around the border of the screen corresponded to the menu items they were next to? *

In the gate and double click methods

- Yes
- No

Did the colours help you remember menu item locations? *

- Yes
- No

Did you find it easier to remember menu items if they were in particular locations? *

Such as corners or the centre of a menu

- Yes
- No

Please explain your answer to the above question *



In the gates method, after a menu item was selected, it briefly flashed on the screen.

Did you notice this? *

- Yes
- No

Did the flash help you remember the locations of menu items? *

- Yes
- No

In the hold-and-click method, did you find that you started to open the menu near the location of the item you were going to select? Or did you tend to open the menu from the same point every time? *

- Opened menu near location of item

- Opened menu from the same point every time
- Other

Were some icons easier to understand or remember than others? *

- Yes
- No

Please explain your answer to the above question *



Did you notice that each menu had a theme? *

- Yes
- No

Did the menu themes help you to search for menu items when first starting a method? *

- Yes
- No

Did the menu themes help you remember item locations when you were more familiar with the item locations? *

- Yes
- No

Did you find any selection method especially fun or novel? *

- Yes
- No

If so, which method was especially fun or novel?

- Click
- Hold-and Click
- Double Click
- Gates

Did having (or not having) a fun or novel method affect your performance? *

- Yes
- No

Any other comments?

