Improving Cascading Menu Selections with Adaptive Activation Areas

by

Erum Tanvir

A Thesis submitted to the Faculty of Graduate Studies of

The University of Manitoba

in partial fulfilment of the requirements of the degree of

Master of Science

Department of Computer Science

University of Manitoba

Winnipeg, Manitoba, Canada

Improving Cascading Menu Selections with Adaptive Activation Areas

By

Erum Tanvir

A Thesis/Practicum submitted to the Faculty of Graduate Studies of The University of

Manitoba in partial fulfillment of the requirement of the degree

Of

Master of Science

ABSTRACT

Cascading menus are some of the most commonly-used widgets in graphical user interface (GUI) systems. Depending upon the number of menu items and the number of submenus, cascading menus may have elongated paths with corner steering, which causes navigation and selection errors. To resolve the corner steering problem, most current cascading menus implement an explicit time delay between the cursor entering or leaving a parent menu item and posting/unposting the associated menu. The objective of this thesis is to design, implement, and evaluate Adaptive Activation-Area Menus (AAMUs), a new technique to improve cascading menu performance by resolving the corner steering and time delay problems. This technique creates a localized triangular activation area between the menu and the child submenu that helps in quick diagonal navigation without imposing any time delay. The AAMU technique improves item selection in comparison to existing techniques and also creates a better user experience with cascading menus.

PUBLICATIONS

Some ideas and figures in this thesis have appeared previously in the following publications by the author

Erum Tanvir, Jonathan Cullen, Pourang Irani and Andy Cockburn. Improving Selection in Cascading Pull-Down Menus. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '08)*, pages 1381–1384., 2008.

There is no question that there is an unseen world. The problem is, how far is it from midtown and how late is it open.

— Woody Allen

ACKNOWLEDGMENTS

First of all, I would like to thank Dr. Pourang Irani for his incredible personal and professional support and guidance. He is a great advisor and mentor. Thank you Pourang.

I thank my family for always being there for me. I am grateful to my parents for their endless prayers and encouragement and my husband for his continuous faith and support. I couldn't have done this without either of you.

Thank you to my collaborators: Dr. Pourang Irani, Jonathan Cullen, Andy Cockburn, I learned so much working with you.

Thank you to my committee members, Dr. Andrea Bunt and Professor Herb Enns, for your time, helpful comments, and support. Also thank you to Dr. Rasit Eskicioglu for chairing my thesis defense.

Thank you to the support staff here at the University of Manitoba, in particular Jeff Durston for his technical support with my experiments.

Thank you to Dr. Irani, for his generous financial support throughout my degree.

And finally, thank you to my fellow students in the HCI lab, including Grant Partridge, Mahtab Nezhadasl, Kang Shi, Sean Gustafson, David McCallum, Ed Mak, Fouad Alallah, Cary Williams and Jonathan Cullen for your support, ideas, and help. You are all amazing. You tried out my experiments and read my work, never complaining but always providing me with good suggestions. Thank you.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

ANOVA Analysis of Variance

WIMP	Window, Icon, Menu, Pointing Device
CTAN	Comprehensive $\TeX$ Archive Network
GUI	Graphical User Interface
HCI	Human-Computer Interaction
IDE	Integrated Development Environment
PDA	Personal Digital Assistant

DEFINITIONS

Cascading menu: A cascading menu or a hierarchical menu is a submenu of choices that are related to the item in the parent menu that invokes the submenu. The presence of a submenu is indicated by an arrow to the right of the parent item that invokes/posts the submenu.

Cascading Item: A cascading item or a parent item is an item that has a child submenu associated with it. The items with no child submenus are referred to as non-cascading item.

Cascading Density: The term describes the number of cascading items in a menu. If the number of cascading items in a menu is higher, the value of cascading density is higher as well and vice versa.

Cascading Levels: A cascading level or depth refers to a menus' position in the hierarchy. The first parent menu is at level one, the first child submenu at level two and so on. The more levels a cascading menu has, the deeper it is.

Post/Invoke/Appear: Used interchangeably for submenu activation. When a parent item is activated by either clicking or hovering the cursor over it, the submenu associated with that item will appear to the right of the parent menu.

Unpost/Revoke/Disappear: Used interchangeably for submenu deactivation. When the cursor leaves an already activated parent item, the submenu associated with that item disappears.

Default Technique: Used for the Microsoft Windows traditional cascading menu. For example the start menu in MS Windows.

1

INTRODUCTION

Menus are an important element of a graphical user interface (GUI) and appear ubiquitously in WIMP (window, icon, menu, pointing device) interfaces. They provide users a convenient means of interaction with the system to select and perform various operations. As software systems become more complex, menus expand in size and thereby affect navigation performance. To make menu navigation more efficient and to categorize the selection process, menus are sometimes designed as cascading menus.

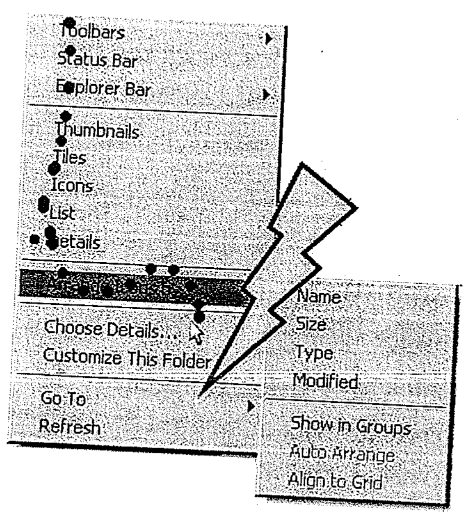

Figure 1: An elongated path causing a movement error. The submenu appears unexpectedly as the cursor crosses the border of the upper item (Adapted from [10]).

Although cascading menus provide the advantage of presenting a large number of selections within a small screen space, they are limited in several ways. In traditional cascading menus, selecting an item in the child submenu requires the user to move the cursor along an elongated path. As a result, menu navigation becomes more difficult with an increasing number of levels in submenus. Users have to slide their cursor through narrow paths causing them to make movement errors since longer and narrower paths decrease efficiency of steering with the mouse or a pointing device [2, 16]. As shown in Figure 1, an elongated and narrow path can cause unexpected selections and unintended submenu appearance or disappearance due to straying mouse movements.

Additionally, traditional cascading menus include a *time delay*. When users are navigating through a menu and bring their cursor to a cascading item, the child submenu is posted after a period of 200 ms. The time delay is intended to improve the steering problem but it slows down the navigation process. An alternate option is to click on the cascading item to open the child submenu to pre-empt the delay. This clicking further slows down the interaction process and over time it can become bothersome for the users. Also, the delay could be too long for some users and too short for others. Additionally, individual preferences depend on many factors, including expertise of the user, context of the operation they are performing, and user fatigue [6].

In my thesis, I have developed a new technique, *Adaptive activation-area menus* or AAMUs, to improve selection and navigation in linear cascading pull-down menus. This technique facilitates the task of target acquisition by removing the need to steer through narrow elongated paths, without forcing the user to wait. This technique introduces an adaptive activation area that changes its size with respect to the size of the child cascading menu and cursor position, providing the user a broad path to steer and allowing diagonal movements. I expect that this technique will enable users to navigate through cascading menus faster and with fewer errors than traditional solutions.

The rest of this thesis is structured as follows: Chapter 2 provides a background that reviews related literature on menus and their various types. Chapter 3 introduces the problem statement that describes the specific problems to be addressed by this research, as well as its scope and limitations. Chapter 4 contains a detailed discussion of the proposed technique, its design and characteristics as well as problems found in the initial design. Section 5 provides a detailed description of the methodology that was used to improve the initial design and the final experiments and their results. Finally, the document ends with summary and future work.

2

RELATED WORK

In this chapter, I will survey the theoretical models specific to menu navigation and selection. I will then describe various traditional techniques in cascading linear pull-down menus and discuss their limitations.

2.1 THEORETICAL MODELS FOR PREDICTING PERFORMANCE IN MENU SELECTION

Researchers have developed theoretical models to predict performance in menu navigation and selection.

Fitts' law: Fitts' law is a robust and widely adopted model for human movement. It was first published by Paul Fitts [8] in 1954. The law predicts the time required to move from a starting position to a final target area and describes the time as a function of the distance to the target and the size of the target.

The mathematical model for Fitts' law and its applications to HCI was established by MacKenzie [12, 13] and is also known as the Shannon formulation. The formulation quantifies the movement task's difficulty, known as the Index of Difficulty or *ID*, in terms of the distance required to capture the target and the size of the target.

$$ID = \log_2\left(\frac{D}{W}+1\right),$$

where D is the distance and W is the width of the object (see [12] for details). The Movement Time or MT is described as:

$$MT = a + b \times ID,$$

where a and b are constants that are empirically determined by linear regression (see [12] for details). Fitts' law predicts that it is easier to capture a target with a large size and is closer to the cursor.

This law has been used to model the action of pointing on computers using fingers and mice and has assisted in designing user interfaces [3, 9, 15]. For example, Fitts' law aided the design of pie menus and resulting studies have shown that pie menus are more efficient and more accurate in comparison to linear menu items [5].

However, Fitts' law has its limitations as well. It applies only to movement in a single dimension and not to movement in two dimensions. Mackenzie and Buxton [14] suggested some changes to improve the model's performance for 2D target acquisition tasks.

Steering Law: Accot's law or the steering law [1] is an extension of Fitts' law to two dimensional modeling and steering of human movement. It predicts the average time necessary to navigate or steer a pointing device (e.g., a mouse or stylus) through a 2D path, tunnel or trajectory. In this path, the user must travel from one end to the other as quickly as possible, while staying within the confines of the path. This law has been used in modeling users performance in navigating a hierarchical cascading menu and it is also used to evaluate the performance of various input devices [2]. This model describes that the time required to travel a trajectory is directly proportional to the distance traveled and inversely proportional to the width of the path. The steering law was mathematically derived from Fitts' law. In its general form, the steering law expresses the time T required to steer through a tunnel as:

$$T = a + b\frac{A}{W},$$

where T is the average time to navigate through the path, W is the width of the path, A is the length of the path and a and b are empirically-determined constants (see [2] for details).

A limitation of the steering law is that the law has been verified for only a few path shapes and widths. For instance, steering is difficult through sharp corners and narrow paths [16], which explains the navigation problems in traditional menus.

2.2 MENU TYPES

Software applications are becoming increasingly complex. More functionality is offered with every new version and, as a result, GUIs are also increasing in complexity. Menus are multiplying in size, making it more difficult for the user to navigate through them. There are various categories of menus for different device types and researchers have developed a number of menu designs for each category to improve menu navigation and the selection process in user interfaces. The main categories include:

Linear Cascading Menus: Linear menus are the most common type of menus in use. They can be used with mice or pens. Menu items are generally arranged in a linear format, listing items from the top to the bottom of the screen or window. The submenus are arranged hierarchically, i.e., a parent cascaded item contains the submenu. The linear cascading menus are further categorized as:

Pull-Down Menus: They are usually used in menu bars, which are located at the top of the window or screen. A user activates the menu by clicking on its name and the menu opens in a drop-down form, presenting the possible operations that could be performed. An example is the menu bar in Microsoft Word.

Pop-Up Menus: A pop-up menu, unlike the drop down menu, can open anywhere on the screen based on the cursor position. An example is the *context menu* in Microsoft Windows, which is activated by right clicking the mouse.

Pen-Based Menus: Pen-based systems allow users to interact using a stylus instead of a traditional keyboard and mouse. Marking menus [11] are an example of pen-based menus.A marking menu allows a user to perform a menu selection by either popping-up a radial (or pie) [5] menu, or by making a straight mark in the direction of the desired menu item without popping-up the menu. Unlike linear menus, marking menus can be operated "eyes free" because selection is based on direction of movement, not position.

Adaptive Menus: Researchers have designed different menu organization schemes for pull-down menus to reduce Fitts' law targeting requirements and to improve performance. Adaptive menus, as the name suggests, dynamically change their appearance or content over time in response to how they are being used. For example, an item list in a menu could be restructured based on usage frequency. Frequently used items are dynamically arranged on the top. Users have no control over the restructuring process. An example of an adaptive menu is the menu bar in the Microsoft Office 2003 suite. Split menus [17] are an example of adaptive menus.

Adaptable Menus: Adaptable menus are user controlled and allow the users to customize the interface on the basis of individual preferences. For example, users can choose the menu items they want to have displayed in the top partition, as well as modify the existing arrangement. A comparison of static, adaptive, and adaptable menus [7] showed that users could optimize their performance if they knew about the possibility of adapting and were able to adapt their menus with a simple interface. Additionally, the results suggested that providing users with control over their menus can lead to better perceived performance and higher overall satisfaction.

2.2.1 *Improvements to Cascading Menus*

Cascading menus are the most commonly-used technique for handling hierarchical menus, however, selecting items from cascading menus is prone to errors. Cascading menus demand a high level of steering accuracy as they require the users to navigate through elongated paths. Also, conventional cascading menus are implemented with an explicit delay for the posting and unposting of the child submenu. This delay makes the selection process very slow. With the increase in complexity and size of cascading menus, there is an increasing demand for improving their design in order to make the navigation and selection process faster and easier. Researchers have designed various techniques to resolve the problems of cascading pull-down menus. Performance improvements have been obtained by either decreasing the distance to the menu items, or by increasing the size of the menu item.

Techniques for decreasing distance

A simple solution to make menu selection and navigation process faster is by reducing the Fitts' Law targeting requirement, i.e., reducing the distance to the target. The steering law also predicts that movement time increases with the length of the path to be covered. Most of the above-mentioned techniques have only focused on the selection of first-level items in cascading pull-down menus. However, longer selection times are caused by steering through long distances, i.e., level two and above. The techniques in Figure 2 have also been tested for higher cascading levels.

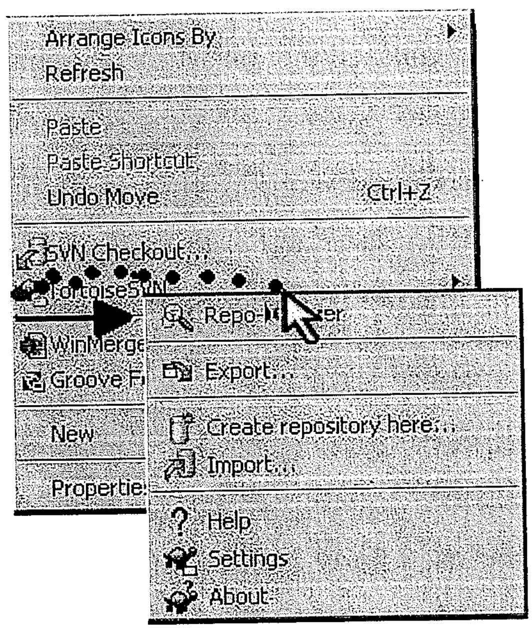

Figure 2: Existing technique: Direction-based cascading menus. Horizontal motion towards right opens a submenu near the cursor position.(Adapted from [10])

Kobayashi and Igarashi [10] presented an improvement to increase the usability of cascading menus by reducing the navigation distance and avoiding the unintended menu posting/unposting. This technique has two components. The first considers the direction of the cursor movement to determine the menu behavior. Vertical movement of the cursor changes the highlighted item within the current menu and the horizontal motion opens and closes the child submenus, therefore, eliminating the unwanted submenu activation during menu navigation. Second, when the horizontal motion occurs, the submenu pops up near the cursor position, hence, reducing the length of the movement path, see Figure 2. A user must move the cursor to the right to open up a submenu or to the left to close the submenu and return to the parent menu.

A user study [10] was conducted to evaluate the performance benefits of direction-based menus over traditional cascading menus. Users were asked to perform a menu selection task. The menu-selection process started with the click of the mouse on a certain item in the menu bar. It ended with the selection of a highlighted menu item. The hierarchical levels of the menus for the above task ranged from two to five. The results of the study showed a 12% decrease in menu selection times as well as 85% fewer unintended submenu activations with direction-based menus.

Although the user study showed that this technique helped in decreasing movement path length, selection time and unexpected submenu activations, there are still limitations. First, the technique adds additional movements to invoke/revoke submenus, which is inconvenient and slows down the interaction process. Every time users need to view a submenu, they have to change the direction of motion, causing them to experience fatigue. Second, as the child submenu opens closer to the cursor position, submenus overlap their parent menus, and hide the rest of the parent menu items. If the user wishes to select a parent menu item while a submenu is open, this overlapping forces the user to make a left horizontal movement to close the submenu first before interacting with the parent menu.

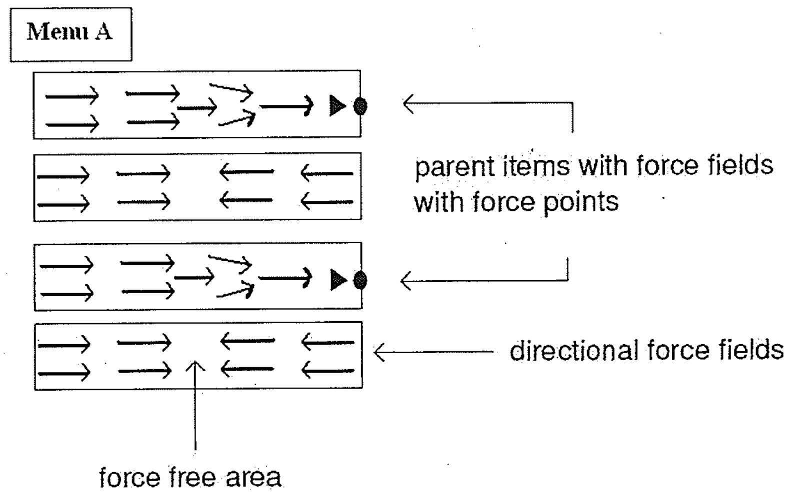

Figure 3: Existing technique: Cascading pull-down menus with force fields (Adapted from [3]).

To make the selection process faster in traditional cascading pull-down menus, Ahlstrom [3] introduced force fields. Force-field menus partially take control of the cursor movement from the users. Two types of force fields are used. First, when moving from left to right within a cascading item, the cursor is pushed towards the child menu and moves faster, optimizing the navigation process. Second, while moving within a non-cascading item, the force fields keep the cursor in the middle of the item, preventing the cursor from falling outside the parent menu; see Figure 3. The most important benefit of force-field menus is that they keep the visual structure of the interface and the interaction technique unchanged.

Ahlstrom [3] conducted a user study to evaluate the performance of force enhanced menus over traditional cascading menus. Users were asked to perform a menu navigation and selection task. The user started the task by clicking a menu and then following the highlighted items. Once the target item was located, selecting the item completed the task. The menu navigation time was recorded. The hierarchical levels of the menus for the above task ranged from two to three. The results showed that the force fields decreased selection times, on average, by 18% when a mouse, a track point, or touch pad was used as an input device.

One disadvantage of this technique is that while moving backwards (from right to left), the users experience resistance due to the force fields acting from left to right. Also, some users do not prefer losing control of the cursor.

Techniques for increasing width

The steering law suggests a second solution for faster steering by increasing the width of the path. A wider path is easier to navigate and less prone to movement errors, causing fewer unintended menu postings and unpostings.

A technique developed by Cockburn and Gin [6] is called *Enlarged activation-area menus* or EMUs, see Figure 4. EMUs improve navigation through cascading menus by increasing the activation area of the parent menu associated with each cascaded submenu, providing a wider path for steering. Also, this technique allows a faster selection

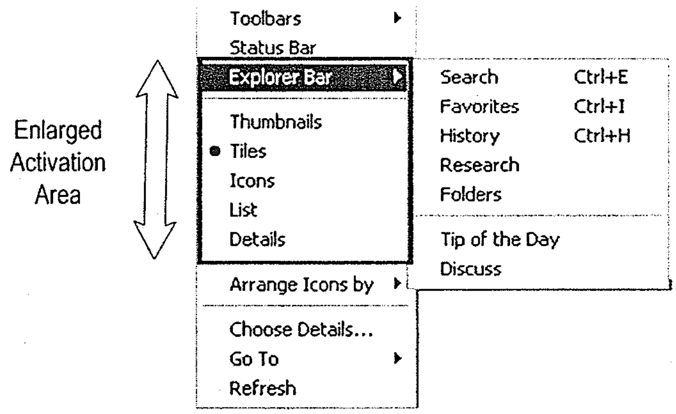

Figure 4: Enlarged activation-area menus (Adapted from [6]).

process by eliminating the problem of time delays. The activation areas for each cascading item are increased by extending them up to the end of the menu or by including all the non-cascading items before the next cascading item.

Cockburn and Gin [6] conducted a user study to compare the performance of EMUs against traditional cascading menus. Users were asked to follow a highlighted path and select the highlighted target. The hierarchy of the menus for the above task was limited to second level menus. The evaluation showed that EMUs allow cascaded items to be selected up to 29% faster than traditional menus.

The problem with this technique is that the activation area is enlarged depending on the density of the cascading items in the parent menu. As a result, in case of adjacent cascading items, the size of the activation area will be equal to that of the traditional cascading menu, offering no performance benefits. Also, users can be distracted when a child cascading menu appears while they are targeting a non-cascading item that lies within the enlarged activation area.

Fitts' Law also predicts that target acquisition can be improved by increasing the size of the target. Fisheye menus [4] and bubble cursors [9] are examples of such techniques.

Fisheye menus [4] dynamically increase the size of the target as the cursor approaches it. They allow many items to be listed on one screen and are a good solution for viewing on small devices like personal digital assistants (PDAs). However, the evaluation of fisheye menus [4] showed them to be slower than traditional cascading menus.

Bubble cursors [9] increase the size of the cursor's activation area at runtime until it encloses at least one target. Bubble cursors are efficient for abstract targeting tasks, such as in computer games where the large cursor area helps in quick capturing of a smaller and fast moving object. However, bubble cursor offers no benefits in discrete targeting tasks where the target item is static and the location is known. An example of such a task is menu selection using cascading pull-down menus.

Being an HCI student, the ultimate goal of my research is to improve the overall experience of computer interaction for users. Therefore, for my thesis research, I have chosen to focus only on cascading pull-down menus which are the most commonly-used type of menu in WIMP (window, icon, menu, pointing device) based interfaces such as MS Windows which is the most used operating system worldwide. I believe any improvement in such a widely used technique will have a significant impact in the field of human-

3

CHARACTERISTICS OF CASCADING MENUS

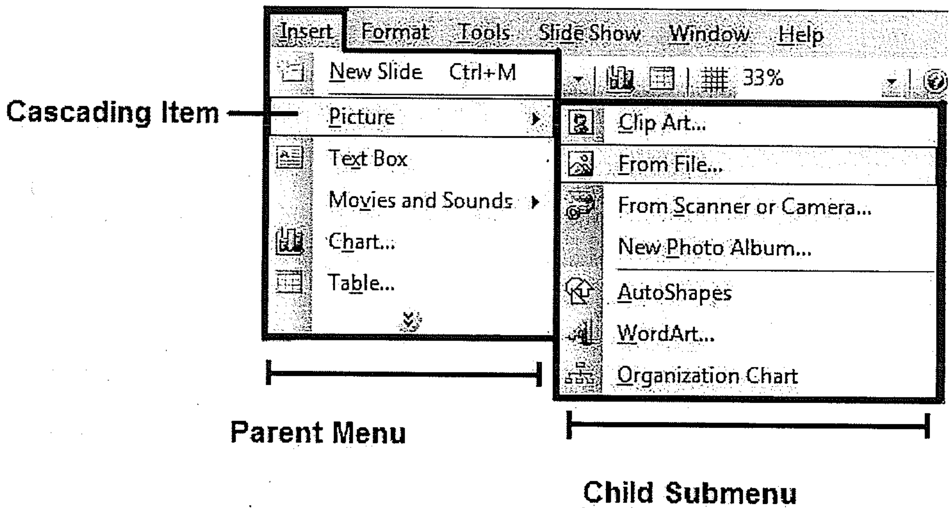

Figure 5: An example of a two-level deep cascading pull-down menu.

A cascading menu or a hierarchical menu (see Figure 5) is a submenu of choices that are related to the item in the parent menu that invokes the submenu. Usually, the presence of a submenu is indicated by an arrow to the right of the parent menu item that invokes/posts the submenu. To invoke a submenu, the user positions the cursor on the target parent item and either clicks or waits for (200 ms) until a submenu appears to the right of the parent menu. The parent menu item that invokes a cascading submenu is also referred to as a *cascading item*. In the example above, "Picture" is the cascading item that invokes or posts a submenu related to picture computer interaction. Other types of menu are outside the scope of my M.Sc. thesis.

manipulation. The *Start menu* in Windows is an example of a very commonly-used cascading menu. Another example is the *Menu bar* in Mac OS.

3.1 PROBLEM STATEMENT

Traditional cascading pull-down menus are the most commonly-used menus in windows based environments. While cascading menus provide an efficient way of organizing a large number of menu items, they are not without problems.

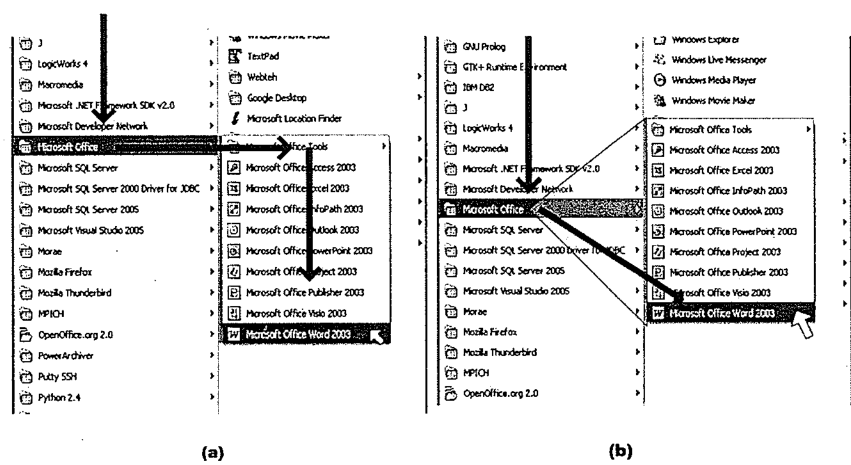

Figure 6: Examples of steering paths: (a) A typical cascading menu with corner steering. (b) A diagonal shortcut path reduces movement time, but includes a time delay.

The first major problem with cascading menus is that they require the user to steer through long and narrow paths. According to the steering law, longer paths take a longer time and cause movement and selection errors [2]. In order to select a menu item in a child cascading menu, users have to move the cursor through at least two sharp corners, see Figure 6 (a). In a study, Pastel [16] found that steering through corners is particularly difficult and slows down the navigation process. Interestingly, paths with a 45 degree corner are negotiated faster than 90 degree corners.

Traditional cascading menus also offer an alternate diagonal shortcut path, but with a time delay. This time delay is too short for some users and too long for others therefore, it not only slows down the pace of interaction, but also cause unintended submenu posting/unposting.

Due to these problems, there is a need for a technique that allows users to steer through broader paths to quickly reach the submenus without any time delay between the submenus. This will enable the users to continue with their interaction process unhindered.

3.2 SUGGESTED SOLUTIONS

Several researchers have investigated these problems and presented two major types of solutions.

Decrease Navigational Distance: As discussed earlier, long and narrow paths make menu navigation difficult and are error prone. Therefore, several researchers have designed cascading menus in such a way that the navigational distance between parent and child menu is minimal. For instance, Kobayashi and Igarashi [10] presented an improvement to cascading menus that reduces the navigation distance by opening the child menu near the cursor position. Although this technique helps in decreasing movement errors, it does not provide any solution for the time delay.

Increase Navigational Area: Corner steering can be eliminated by increasing the width of the navigational path. Diagonal steering can reduce navigational distance as well as the number of errors. There is no need for a time delay in such cases. An example of this design is a technique developed by Cockburn and Gin [6] called *Enlarged activation-area menus* or EMUs. EMUs eliminate the problem of time delay by increasing the activation area of the cascading item inside the parent menu. The problem with this technique is that the activation area is enlarged depending on the density of the cascading items in the parent menu. Therefore, in the worst-case scenario, the size of the activation area will be equal to that of the traditional cascading menu, offering no performance benefits.

3.3 DESIGN CRITERIA

Based on the above summary, I propose that an efficient cascading menu design should fulfill the following criteria;

1. A user should be able to select an item in a child cascading menu (or submenu) without activating any other cascading items in the parent menu.
2. A user should be able to trigger any of the parent menu items while a child cascading menu is activated.
3. Time delays should be minimized or avoided if possible.

4. The selection process should be efficient, i.e., faster and more accurate than the default technique.

None of the existing techniques fulfill all of the above criteria.

4

DESIGN AND EVALUATION OF AAMUS

In this chapter, I describe the design and evaluation of a new linear cascading pull-down menu technique called *Adaptive Activation Area Menus or AAMUs.* I specifically designed AAMUs to overcome some of the problems with existing cascading menu techniques outlined in chapter 3.

4.1 THE AAMU DESIGN

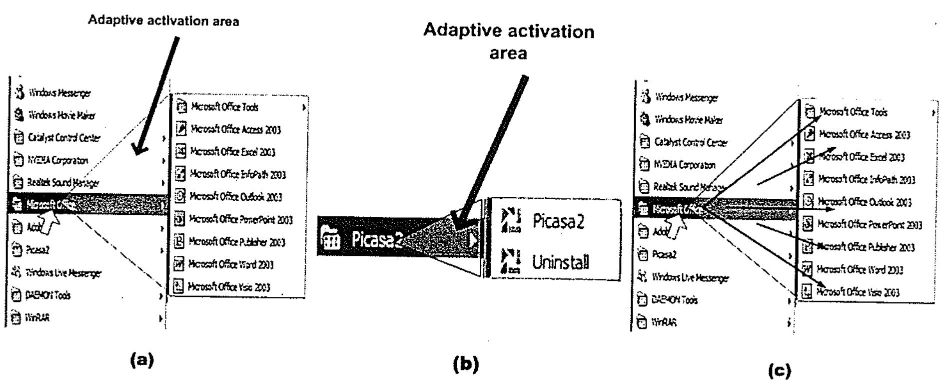

Figure 7: Examples of adaptive activation areas: (a) A long child cascading menu. (b) A small child cascading menu. (c) Diagonal steering paths.

4.1.1 *Characteristics*

AAMUs provide users with a broad steering path by means of adaptive activation areas. An *adaptive activation area* suggests that the size of the activation area is dynamically determined depending on the size of the child cascading menu and position of the cursor, see Figure 7. This activation area is triangular in shape and overlaps some area of the adjacent menu items. The activation area is semitransparent allowing the users to see the rest of the menu items. Also, the broader activation area provides a means to remove the time delay before a cascading submenu is posted, because the activation area removes the ambiguity of the user's intentions.

As an AAMU adapts to the size of the child submenu and initial cursor position, there are two different submenu alignments possible.

Center-Aligned: If the size of the child menu permits, i.e., if there is enough space available at the top of the cascading item, then the child submenu is placed such that half of its height is above and half is below the cascading item. Hence, the name center-aligned, see Figure 8 (a).

Top-Aligned: If the child submenu is too long to be placed centrally, then its top is aligned with the top of cascading item. Hence, the name top-aligned, see Figure 8 (b).

4.1.2 *The Technique*

The AAMU technique works as follows:

- During navigation, when the users rest the cursor on a cascading item, a transparent activation area is invoked along with the child submenu.
- This activation area adapts its size and starting position according to the size of the child menu and the initial cursor position.
- To chose a submenu item, users can move diagonally towards the child cascading menu. The child submenu remains posted as long as the cursor remains inside the triangular activation area, see Figure 7 (b).
- The activation area remains present as long as the cursor is inside the area. If the users want to activate another item, first they have to move the cursor outside the boundaries of the current activation area and then point to the desired item.
- As soon as the cursor enters into another cascading item in the parent menu, the previous activation area disappears and the new activation area and child submenu appear without any delay.

I have also designed a variant of AAMU by combining force fields and AAMUs. This combination technique is called force-AAMU. Force-AAMUs provide an additional benefit of reduced navigation distance in addition to wide steering paths. Force fields are only implemented within the adaptive activation area. Once the cursor enters the activation area, it is pushed towards the right side. As there are no force fields in the menu items, no resistance is experienced while entering back into a parent menu, unlike in a force fields menu.

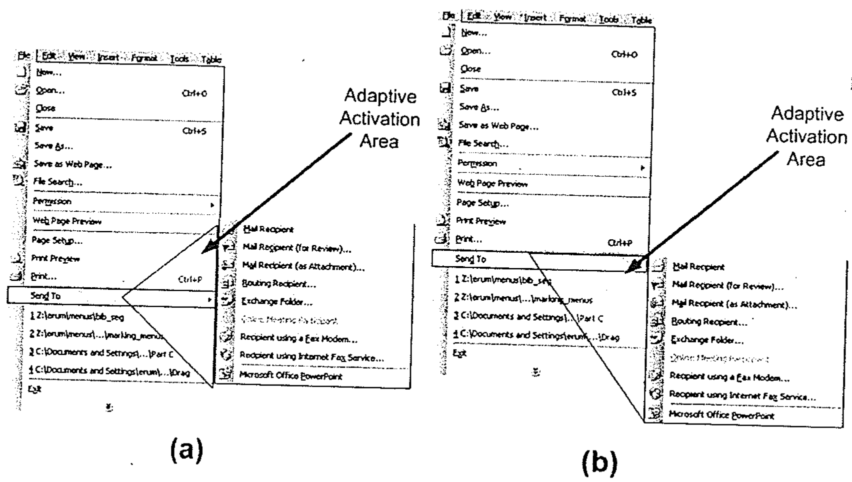

Figure 8: Examples of different alignments for AAMUs (a) A centrally-aligned child cascading menu. (b) A top-aligned child cascading menu.

Both techniques are implemented in Microsoft visual studio .NET, using C# as the programming language.

4.2 ADVANTAGES OF AAMUS

1. The broader activation area enables users to navigate without turning sharp corners, which also reduces the overall navigation distance.

2. This technique reduces selection errors and fulfills the first criteria by permitting diagonal movements without deactivating the child submenu.

3. The adaptive activation area increases and decreases its size with respect to the size of the child cascading submenus, offering performance benefits in all types of scenarios.

4. The semitransparent activation area enables users to continue interacting with the parent menu while the child submenu is active, fulfilling the second design criteria.

5. Removing the time delay when opening a submenu improves the overall selection time.

6. Force-AAMUs further reduce selection times by reducing the navigation distance.

4.3 USER STUDIES TO EVALUATE AAMUS

To validate my menu design I ran a user study (this study and its results have been published [18]). The study compared AAMU and its variant force-AAMU, with other existing techniques, including traditional cascading menus (default), gesture-based menus [10], enlarged activation area menus (EMUs) [6] and force-fields [3]. For the user study, all menu types were implemented without any time delay. I compared AAMUs against the strongest existing cascading menu designs to provide a fair perspective into the merits of each technique.

4.3.1 *Experiment1: Selection*

Method

The experiment was conducted on Windows XP using a Pentium 4 machine with 1 GB of RAM. The experiment was performed using a mouse.

Participants: Eleven university undergraduate students participated in exchange for course credit. All of them had used the MS Windows default menu and were familiar with operating a mouse. None were color blind.

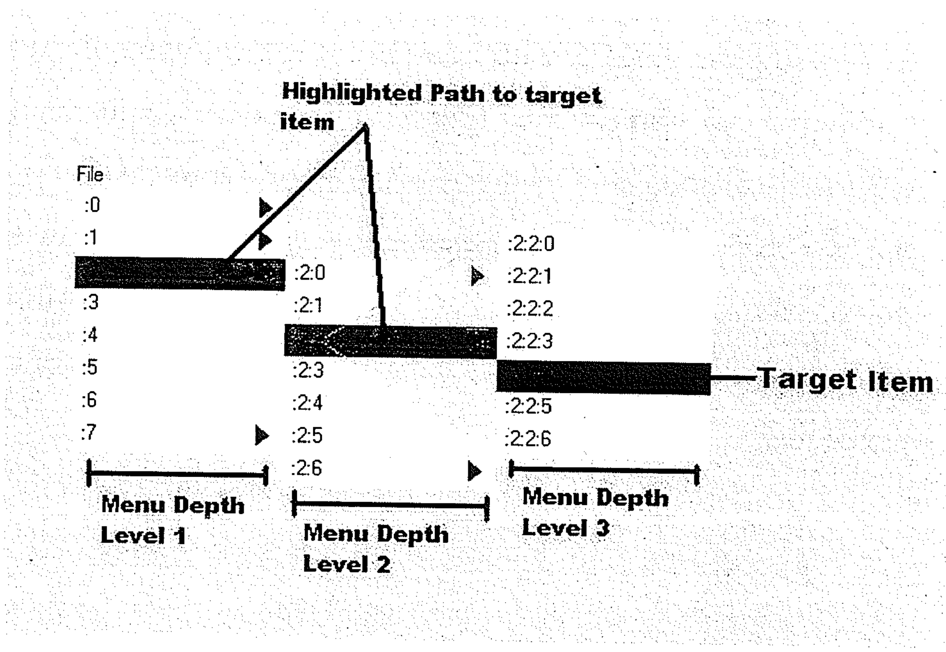

Figure 9: An example of a 3-level-deep selection task in experiment1. The red item is the target and the green items indicate the path.

Menu Types: The following menu types were tested in this study: default, AAMU, force-AAMU, EMU, force-fields and gesture-based.

Task and Stimuli: Participants were required to perform 30 menu selection tasks with each technique, with 10 trials at each of three cascading menu depths (2, 3, and 4). As in a typical selection task, the user always know the location of a target item therefore, the path to the target menu item was highlighted in green to provide users with a visual cue (see Figure 9). The target menu item that users had to select was displayed in red. Menu length was varied randomly in each level of depth in every trial with a constant cascading density of 50%. The target menu item always appeared in the last menu depth level. For each trial, a different path and target position was randomly generated to prevent users from learning the trial path and positioning of the target item. At the start of the experiment, the participants were given five minutes of training with each menu type. Participants were instructed to complete tasks as quickly and as accurately as possible. The order of presentation was first controlled for menu type and then for depth such that 30 consecutive trials for each menu type with random depths were presented at a time. For presentation sequence, a Latin square of value 6 was used for 11 participants. With 6 menu types, 3 depths, and 10 trials per condition, the system recorded a total of 180 trials for each participant. Participants were allowed breaks between trials. The experiment took approximately 25 minutes.

Design: The logged dependent variable (task time) was analyzed using a 6×3 repeated measures analysis of variance (ANOVA) for factors interface type (default, AAMU, EMU, force-fields, force-AAMU, and gesture-based) and menu depth (targets at cascade depth 2, 3, or 4).

Results

The overall results for speed are shown in Figure 10 and the raw statistics are available in Appendix A.1.

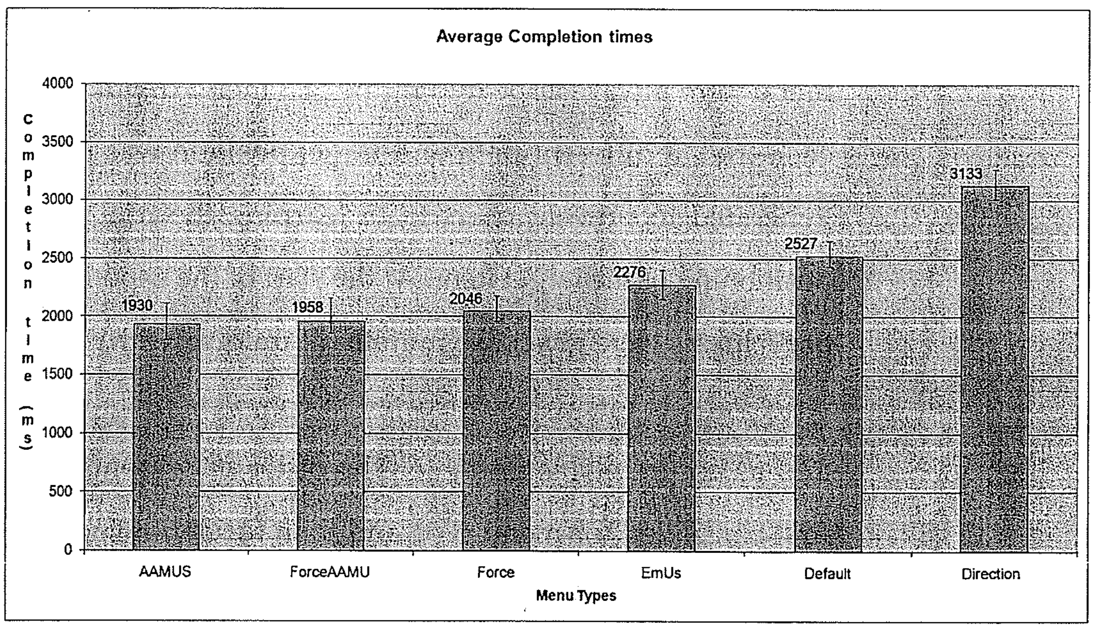

Figure 10: A bar graph showing average completion time for different menu types in experiment1.

There was a significant main effect of menu type on speed ($F(5,50)=28.5$, $p<.001$). Additionally, as expected, there was a significant main effect for depth on speed ($F(2,20)=172.4$, $p<.001$). There was also a significant menu type×depth interaction effect ($F(10,100)=8.9$, $p<.001$). The interaction graph is shown in Figure 11. The cause of the interaction is apparent in the figure, with performance degrading much more rapidly across depth with default and gesture behaviors than with the other interfaces. The raw statistics are available in Appendix A.1.

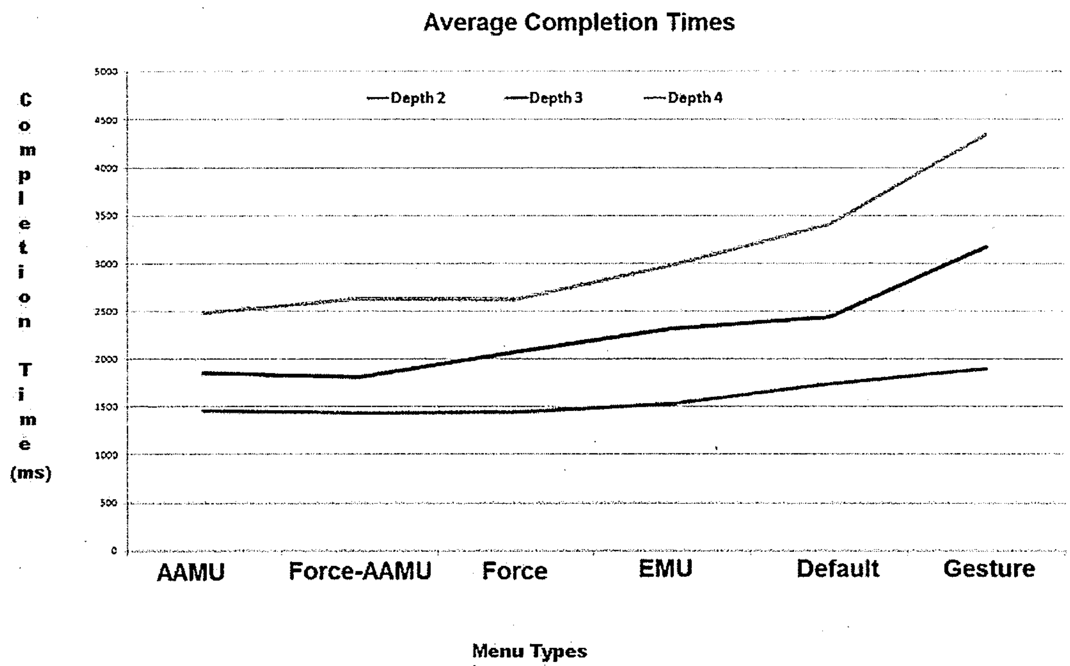

Figure 11: An interaction graph showing rapid decrease in speed with increase in depth, for different menu types in experiment1.

To examine performances of individual menus, I compared each possible pair of menu types against each other. This was done by computing a Tukey post-hoc pairwise comparison, using a Bonferroni adjustment. The comparison showed that the top 3 performing techniques were AAMUs, force-AAMUs and force-fields. AAMU (mean 1.93s, sd 0.55) and force-AAMU (1.95s, 0.57) were significantly faster than EMUs (2.28s, 0.75), default (2.53s, 0.91) and gesture-based (3.13s, 1.36). Force-fields (2.04s, 0.62) was only significantly better than default. However, there was no significant difference between AAMU, force-AAMU and force-fields. Also, gesture-based was significantly slower than all other menu types. The post-hoc comparisons are summarized in Appendix A.1.

Subjective Rankings

A post-study questionnaire was collected from participants asking for their most preferred technique. The questionnaire is available in Appendix A.1. AAMU was the most preferred technique, followed by force-AAMU, EMU, Force fields and Gesture based. Overall preference leaned towards any technique that implemented an enlarged activation area, which is a common feature between AAMUs and EMUs (see Figure 12). Users gave lower preference to EMUs due to the non-uniform activation area which was distracting and confusing. Those who did not prefer force field menus commented that they were more familiar with the standard speed of the mouse. The increased cursor acceleration, due to force fields, made it feel as if the control was taken away. The majority of the users disliked the gesture-based menu on the basis that it interfered with the pace of interaction by forcing the user to change their direction of motion during the posting/un-posting invocations.

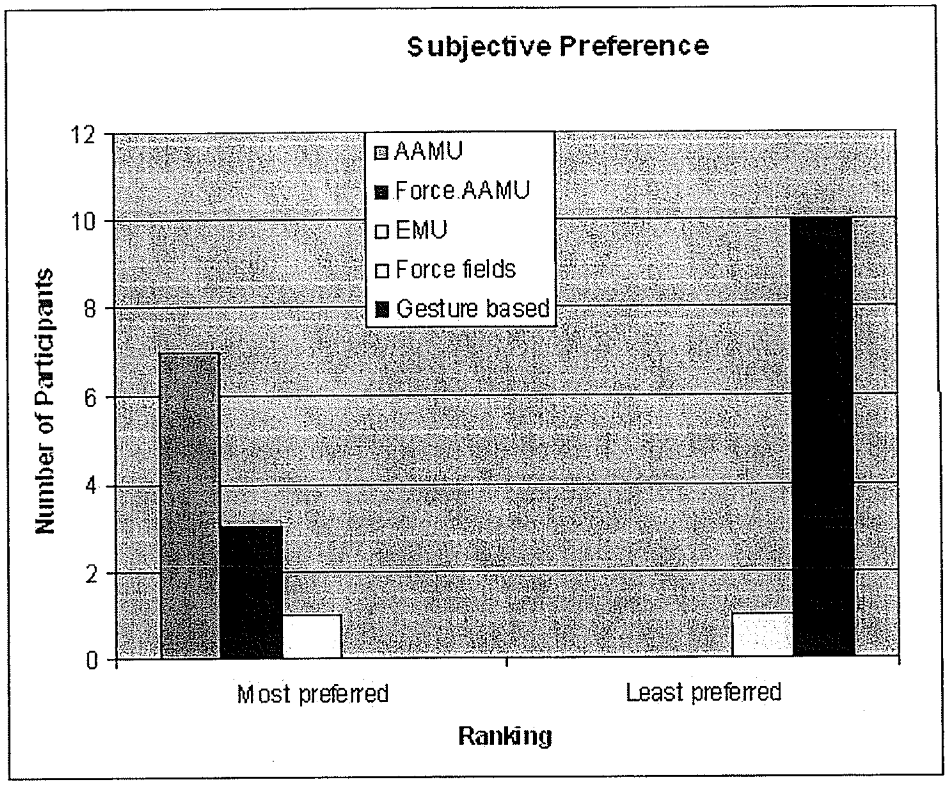

Figure 12: A bar graph showing subjective preference for different menu types in experiment1.

4.3.2 *Experiment2: Search*

The above user study was performed on the selection task only. In most real world scenarios, users also perform search tasks along with selection. Searching not only takes more time but is more error prone due to the frequency of in and out of menu movements. Therefore, I wanted to evaluate the performance benefits of AAMU in search tasks as well. Furthermore, I had only evaluated AAMU with the mouse. It was important to evaluate this technique across a range of popular devices such as the touch pad. Therefore, I ran another user study with three input devices and a search task.

Method

This experiment was also conducted on Windows XP using a Pentium 4 machine with 1 GB of RAM. The experiment was performed using a mouse, touch pad and a stylus. The experiment was designed as between subjects and all participants used all three devices.

Participants: Twenty university undergraduate students participated for course credit. All of them had used the MS Windows default menu and were familiar with operating a mouse. Ten of the participants had prior experience using a touch pad and only three of them have used a stylus before. None were color blind.

Menu Types: Based on poor performance and lower subjective rankings in selection study, gesture-based [10] and force-fields [3] were dropped from this study. Also, due to the lack of support for stylus based input in force enhanced menus, force-fields and force-AAMU were not included. Following menu types were tested in this study: default, AAMU, and EMU.

Task and Stimuli: Participants were required to perform 20 menu search tasks with each technique, at a fixed menu depth (level 3). Since it was a search task, no visual cue was provided for the path and participants had to activate all cascaded items to search for the target. The target menu item was displayed in red. Menu length was varied randomly in each level of depth in every trial with a constant cascading density of 50%. The positioning of the target item was determined randomly but, always appeared in the last menu depth level. For each trial a different path and target were generated to prevent users from learning the trial path and positioning of the target item. At the start of the experiment, the participants were given five minutes of training with each menu type. Participants were instructed to complete tasks as quickly and as accurately as possible. The order of presentation was first controlled for menu type and then for depth such that 20 consecutive trials for each menu type with random depths were presented at a time. For presentation sequence, a Latin square of value 3 was used for 20 participants. With 3 menu types, 1 depth level, 3 devices and 20 trials per condition, the system recorded a total of 180 trials for each participant. The experiment took approximately 25 minutes per participant.

Design: The logged dependent variable (task time) was analyzed using a 3×3 repeated measures analysis of variance (ANOVA) for factors interface type (Default, AAMU and EMU) and input devices (mouse, touch pad and stylus).

Results

The mean completion times with respect to menu types and device types are summarized in (Figure 13) and the raw statistics are available in Appendix A.2.

There was a significant main effect of menu type ($F(2,36)=11.668$, $p=.001$) on completion time. There was also a significant main effect for device types ($F(2,36)=63.02$, $p<.001$) on completion time. However

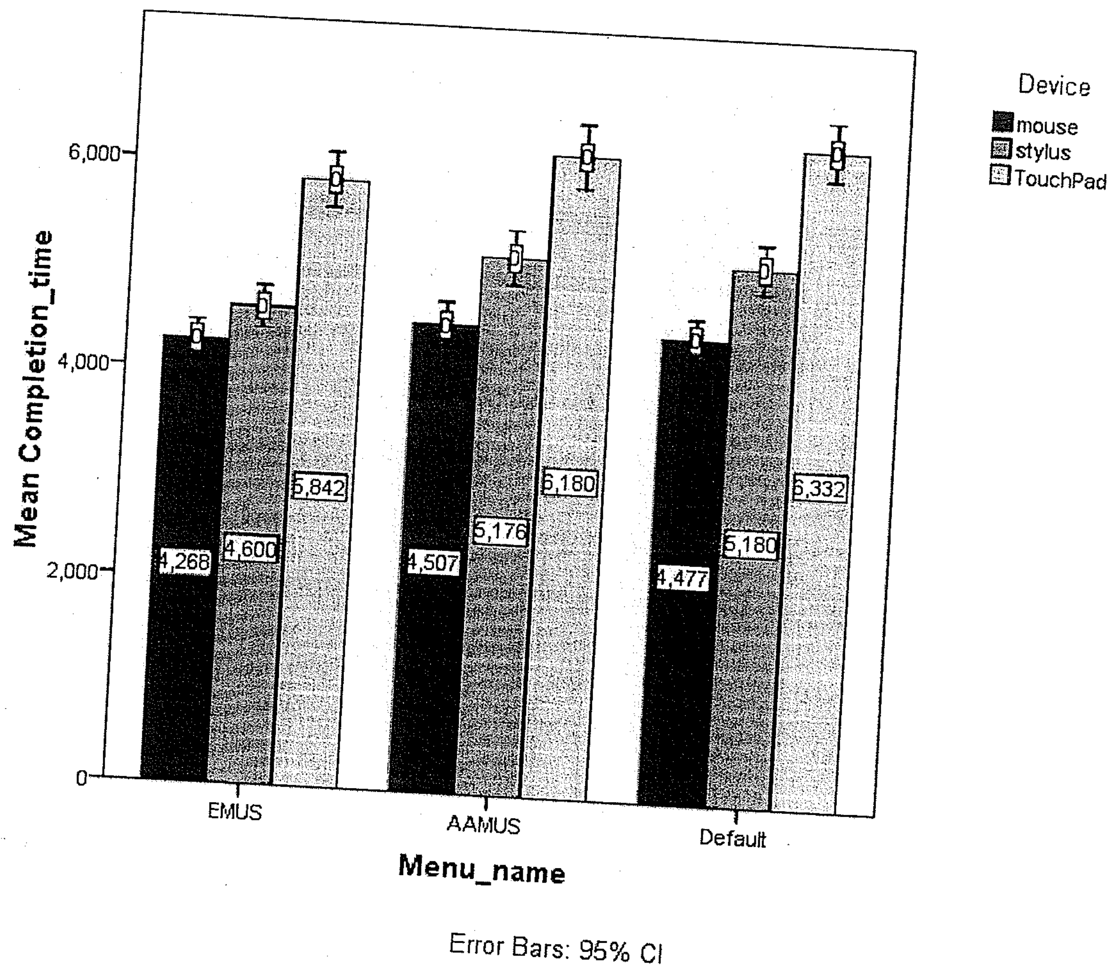

Figure 13: A bar graph showing mean completion times with respect to device types in experiment2.

there was no significance found between menu type×device types interaction ($F(4,72)=1.510$, $p=.208$).

An overall pairwise comparison, using Bonferroni adjustment, of menu types alone showed EMUs performing significantly faster than default and AAMUs (p<.001 and p=.006) where as there was almost no difference among default and AAMUs. However I also wanted to see how each menu type performed with each device. Therefore,

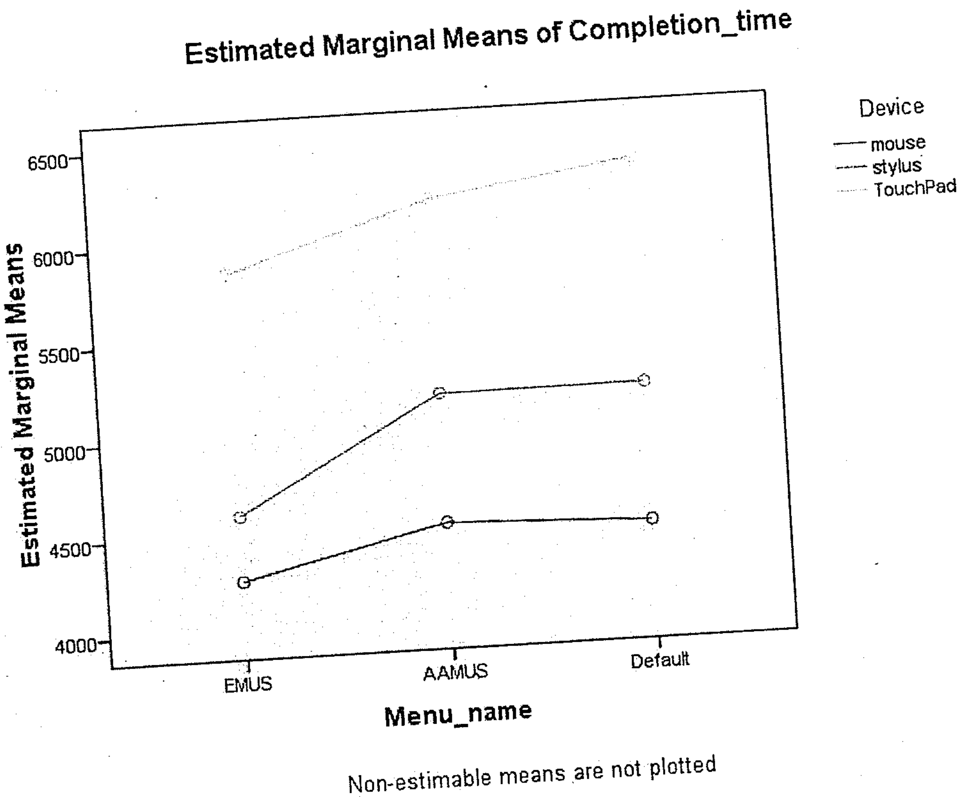

Figure 14: An interaction graph showing significant performance degradation in AAMU and default techniques in stylus device only.

to examine menu type and device interaction effects in more detail, I compared each possible pair of menu types for each device type. This was done by computing post-hoc pairwise comparisons, using Bonferroni adjustment, in the Univariate ANOVA. The comparisons showed that despite supporting lower completion times in all three devices, EMUs (mean 4.87s, sd 2.30) were only significantly faster than both AAMUs (5.25s, 2.75) and default (5.29s, 2.4) with the stylus ($p=.002$ and $p=.001$ respectively). EMUs were also significantly faster than default technique, in touch pad ($p=.038$). However there was no significant difference between the three menu types on the mouse. Also, there was almost no difference between AAMU and default on all three devices. The device and menu types interactions are shown in Figure 14.

4.4 DISCUSSION OF RESULTS

The above-discussed studies were only preliminary tests to evaluate the potential of AAMU technique. I tested AAMU against item searching and selection tasks and saw promising results. These experiments not only showed the benefits of the AAMU technique but also pointed towards some potential design problems.

Item Selection, in menus, is the most commonly performed task and the results showed AAMUs with the lowest task completion times against all other menu types, and were significantly faster than default, EMUs and gesture based ($p=.003$, $p=.004$ and $p<.001$ respectively). Force-AAMUs were the next fastest menu type being significantly faster than default, EMUs and gesture based ($p=.011$, $p=.028$ and $p=.001$ respectively).

Item searching is a more time consuming task than selection. Also, as users are exploring different menus in search of the desired item, they are more prone to make movement errors. Although there was no significant loss of performance, however the results clearly indicate that AAMUs were not as efficient in searching task as they were in selection. One possible explanation for the increased task completion time is that the wider activation area makes the process of exploring menus very slow. When the wider activation area is fully expanded, it covers adjacent items and the user cannot activate the adjacent menu item immediately. I refer to this limitation as the "cursor trapping" problem since the user needs to "get out" of the activation area before entering the next menu item. Another reason is that this technique introduces an enlarged activation area and users have to adjust and re-familiarize themselves with the new interface.the wider activation area of AAMUs makes menu exploration difficult.

5

IMPROVING AAMU PERFORMANCE

The existence of the aforementioned problems pressed for the need for further investigation. My aim in this part of the research was to conduct more studies to pin point the design problems in AAMU and improve the technique to make it better than the conventional menu. Following are the factors that I studied for improving the existing design:

Utilization of Diagonal Path: I wanted to know if users were using the diagonal AAMU path. This would justify the choice of a triangular-shaped activation area. From my observations, I learned that every user used the diagonal path in almost every trial.

Users' Movement Patterns: I was interested in learning the users' movement patterns as they could either vary depending on individual user or with respect to the relative positioning of the child item. I was hoping that observing these patterns would assist me identifying wether "cursor trapping" was the real problem or not. Also, it could point to some other problems which were yet unknown. I learned that users had almost identical patterns based on child relative position unless they were trapped. I also found that cursor trapping was the real cause of slow navigation times.

5.1 RESEARCH QUESTIONS

- Identify the source of the trapping problem. What exactly is causing the trapping problem? Is it related to user movements and does it differ on a user-by-user basis or is there a common pattern among all users that is causing it?
- If I design AAMUs to eliminate the trapping problem, will it improve performance in other tasks, such as item searching?
- If cursor trapping is not the problem then, will I be able to identify any other problems, in the AAMU technique, by observing users' movement pattern?

5.2 METHODOLOGY

The following is the methodology I used for addressing these questions.

Observations

The first step towards gaining insight into the trapping problem was to observe how people interact with AAMUs. I designed a setup in which users performed menu navigation and selection tasks using AAMUs. The interaction and navigation patterns of the users and other important values, like number of clicks and performance times, were logged. These observations provided me with an account of the problems faced by users while interacting with AAMUs. The knowledge gathered by these observations also helped me identify the steering patterns of users while activating and entering submenus. The results of the observation study helped me improve the AAMU technique. The observation study and its findings are described in section 5.3.1.

Design and Implementation of the Techniques

After the observations, the next step was to design the improved techniques based on the analysis of the results. I considered the following factors while designing the new techniques.

Shape: Various shapes of the activation area with respect to the common steering pattern of users.

Time Delay: Different values of time delay in posting/unposting the child submenu.

Drawing Position of the Activation Area: Activation area can be drawn a few pixels ahead or behind the cursor.

Visual Cues: Provision of some form of visual cues to help users get out of the trap.

Using the above mentioned factors, multiple designs of the new AAMUs were created. Those designs are discussed in detail in section 5.3.2.

Evaluation of the Improved Techniques

Once the new designs were implemented, the next step was to test them for problems. Also, the new designs needed to be evaluated against each other as well as against the default technique to find the best performers among them. The experiment and its findings are described in detail in section 5.3.3.

Based on the results from experiment 2, the weaker designs were eliminated and the stronger designs were further improved . The new improved designs were then put to the final test against the default technique. This experiments and its findings are described in section 5.3.4.

5.3 DESIGN AND IMPLEMENTATION OF THE IMPROVED TECHNIQUE

In this section I will describe in detail every step of the Methodology discussed above.

5.3.1 *Experiment3: Data Collection and Testing for trapping*

This experiment aimed at finding out if trapping was the cause of long navigation times, as well as to identify common user navigation patterns. In this experiment, the menu navigation and selection tasks were designed such that half of the tasks had no chances of trapping (clear case) and half of them essentially had trapping (trapped case). The hypothesis was that if AAMUs performed better in the clear case as compared to the trapped case, it would support the assumption that cursor trapping was a major problem with AAMUs.

Data Recorded:

- **Position of the mouse:** All x and y coordinates from the moment the cursor enters the menu till the end of the task.

- **Position of the parent item:** The position of all parent and target items in the task.
- **Trapping:** If trapping occurred.
- **Completion time:** Time in milliseconds from the moment the user activates the menu until the target is clicked.

Method: The experiment was conducted on Windows XP using a Pentium 4 machine with 1 GB of RAM. The experiment was performed using a mouse.

Participants: Nine university undergraduate students participated for exchange of course credit. All of them had used the MS Windows default menu and were familiar with operating a mouse. None were color blind.

Task:

- The experiment was conducted using both search and selection tasks.
- In the search task, users were required to browse all the parent items (in other words, all possible paths) until the target item was located and a single click on the target item completed the task. No visual cues are provided for the path. The target item in both tasks was highlighted in red.
- In the selection task, the path to be followed was highlighted in green and the user would follow the path until the target item is located and a single click on the target item completes the task.

- Each task was two levels deep. The second level had fifteen items. The length of the first level varied between four and eight items randomly.

- In the trapped case, there were two adjacent parent items located to ensure trapping. However, in the clear case, only one parent item within a menu was placed so that no trapping can occur.

- The menus were drawn in the center of the screen to enable center alignment in all scenarios.

In order to observe all possible navigation paths, from parent menu to the child menu, five relative target positions were tested. Each position was two items far apart from the previous and next position, hence, the suitable menu length was fifteen items long. The five relative positions were 2, 5, 8, 11 and 14 (see Figure 15)

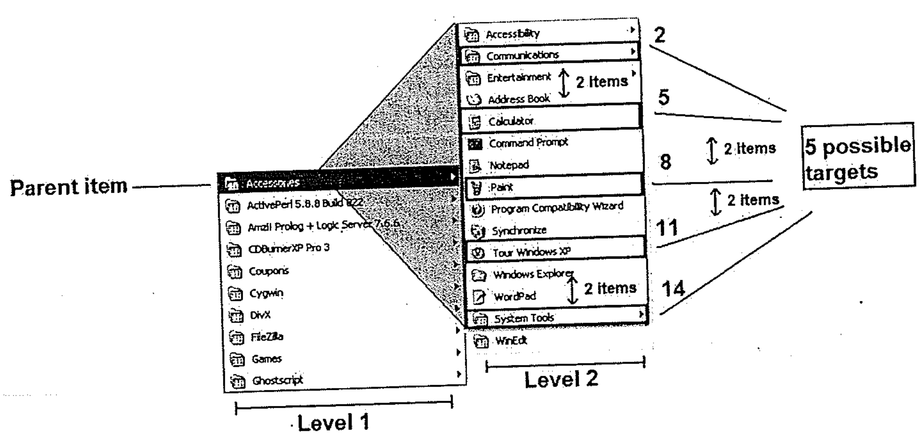

Figure 15: A two-level deep menu showing five possible target positions. Each target position is two items apart.

Besides providing a means for learning user patterns, this study also aimed at finding if AAMUs were significantly better than the default technique when there was no trapping. Therefore, in both type of tasks and cases, AAMUs' performance was compared against the default technique. If the performance of AAMUs in the clear case was significantly better than the defaults' performance in the clear case, it would serve as an evidence that the improved AAMUs will also be significantly better than the default technique.

Design of the experiment: For depth level two, there were five possible targets, hence, the total number of possible paths was five. There were three trials per task and ten trials for practise. In case of trapped tasks each path was repeated twice to account for two parent items. The experiment design was $2 \times 5 \times 2 \times 2$ within-participant design. (2 menu types, 5 paths, 2 testing conditions for trapping, 2 task types). The experiment was counterbalanced using a Latin square to eliminate the bias for menu types.

Evaluation of the Data: The tasks with the active trapping flag were evaluated for the trapped case and tasks without the active trapping flag for the clear case. Although this did not provide an equal number of tasks for both the clear and trapped scenarios, my aim was to collect as much data as possible.

The results were analyzed by averaging the time taken by all the participants to complete each task and by averaging the number of tasks. A paired sample t-test was used to compare the completion time of AAMUs with the default technique and the statistical significance was measured(at $p<0.05$ level).

Findings of Experiment3

The trials were categorized as trapped and clear based on the trapping flag. Each category was then analyzed using a paired sample t-test.

As expected, the analysis of all clear trials showed AAMUs (mean 1.28s, sd 0.193) performed significantly better than default (1.53, 0.27), i.e., ($t(9)$=-4.890,p=.001). Whereas, in the case of trapped trials AAMUs(1.64, 0.3) were slightly faster but not significantly better than default (1.71, 0.27), i.e., ($t(9)$=-0.642,p=0.539). Hence it confirmed the hypothesis that the trapping problem is the real cause for AAMUs' poor performance. The raw statistics are available in Appendix A.3.

I also recorded the user navigation paths for each individual user per trial to observe if there were any common patterns to be found based on the target location and trapped cases. If I could identify some distinct navigation pattern among all users in case of trapping, it would help me in creating more efficient designs for AAMUs. Based on my observation I divided all trials into three categories:

No Trapping: In case of clear trials there was no trapping. When I observed the steering patterns, it was evident from the graphs that users faced no problem reaching their target item. Almost all users showed identical navigation patterns in case of clear trials by making use of the broader activation area and performing diagonal steering, see Figure 16. This also confirmed my hypothesis that the broader navigation area of AAMUs would help users in improving the menu navigation.

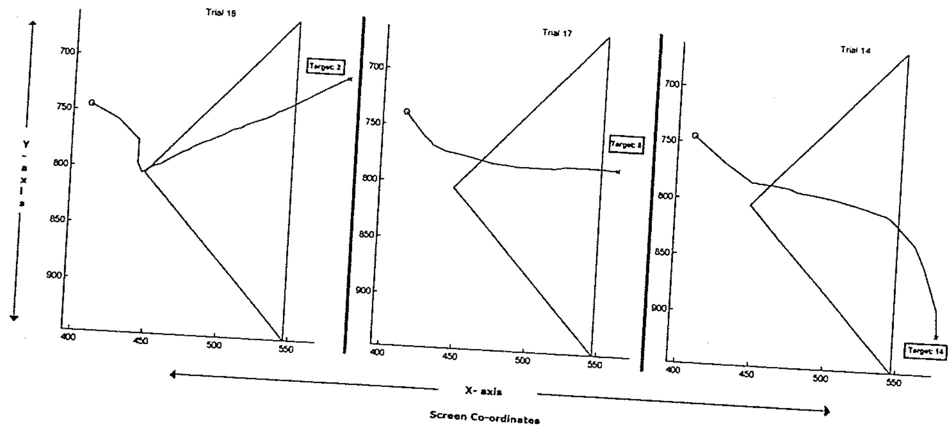

Figure 16: A collection of three clear trials with three different target positions at top(2), center(8) and bottom(14). In all the cases, we see the path moving within the AAMU. Also, users utilized the diagonal path with respect to target position.

Trapping: In case of trials designed for trapping, users were shown menus containing adjacent parent items making it highly likely to cause trapping. If trapping actually happened then that particular trial was flagged as a "trapped trial". As expected, majority of these trials actually caused cursor trapping and users had to maneuver the cursor out of the AAMU triangle to select the correct item. In the graphs, the navigation patterns showed clearly how much trouble was caused by cursor trapping. There were all kinds of patterns visible from back tracking to extreme vertical (upward or downward) cursor movements.

In figure 17, the graph depicts a user's cursor movement pattern in a trapped trial. There were two adjacent parent items, item 2 and item 3. Item 2 was the false parent and item 3 led to the target item that was located at the second position in the child submenu. The user started moving downwards in the parent menu and when entered inside item 2 at "pt 1" (see Figure 17), the first AAMU triangle was activated but it was not the desired item so the user

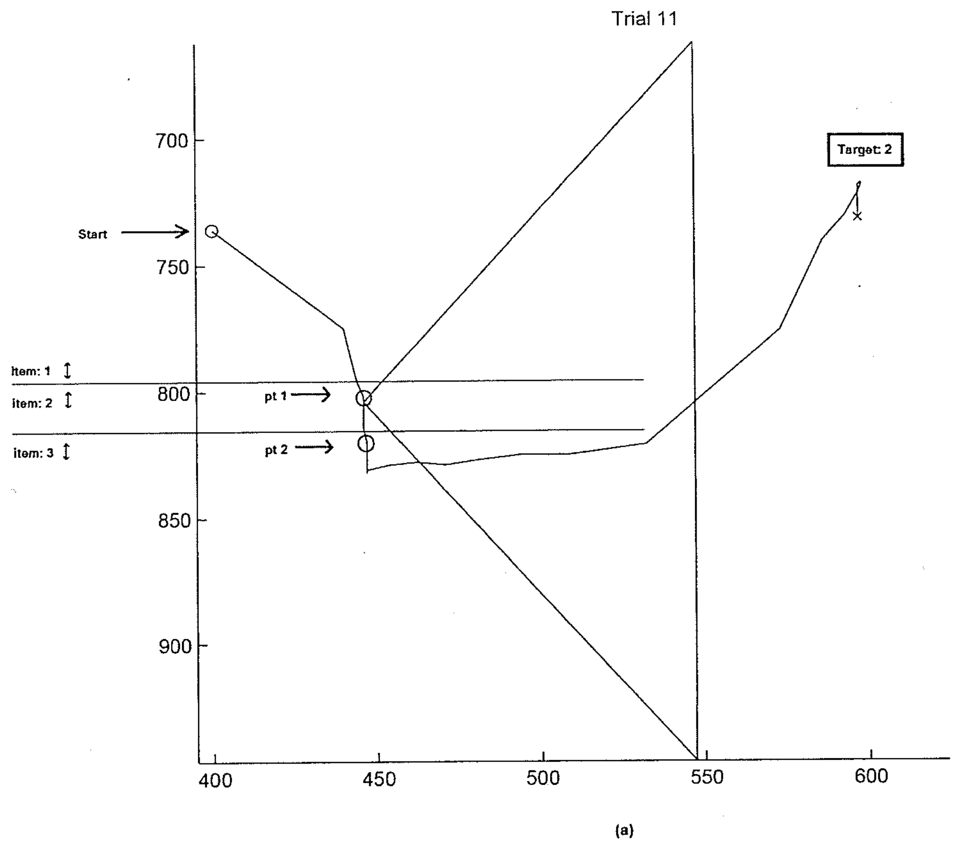

Figure 17: A graph showing navigation patterns in a trapped trial. The user got trapped at "pt 1" and kept moving downwards until the desired item was activated at "pt 2".

kept moving downwards and activated the second AAMU at "pt 2" which lead to the target item. This is an example of a trapped trial with minimum negative impact. Since the users' motion was vertical rather than diagonal, it was easier to get out of the trap quickly.

In figure 18, another navigation pattern in a trapped trial is displayed. In this trial the two adjacent parent items were item 0 and item 1 whereas item 0 was the false parent and item 1 led to the target item that was located at position 5 in the child submenu. This graph shows how much interruption could be caused by cursor trapping. As seen in Figure 18, the user started moving downwards in the parent menu and as soon as the cursor entered the boundaries

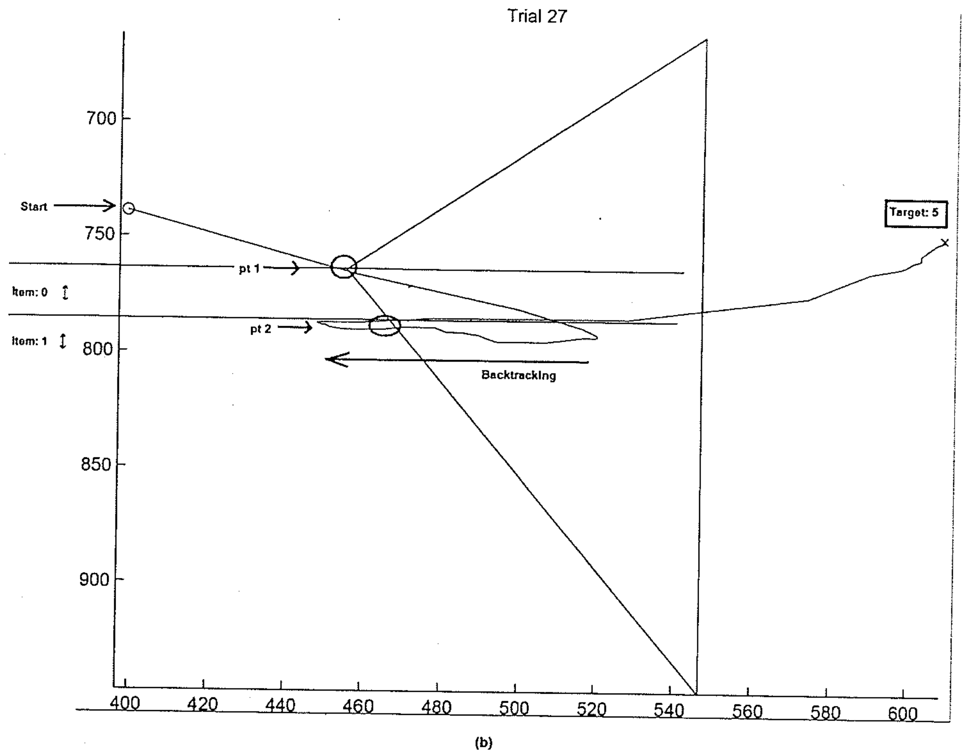

Figure 18: A graph showing backtracking of cursor in a trapped trial. The user got trapped at "pt 1" and had to move all the way back to get out of the wrong AAMU and activate the correct AAMU at "pt 2".

of item 0, at "pt 1", the respective AAMU was activated. Since the users' motion was diagonal, the cursor moved much inside the triangle before the user realized that it was the wrong item and now the only option left was to move the cursor outside the triangle to deactivate it. So the user back tracked all the way out of the AAMU triangle and as soon as the cursor entered the boundaries of item 1, at "pt 2", the other AAMU activated and it lead to the target item. By observing this graph, I realized that if the user had a choice of somehow disabling the wrong AAMU and enabling the desired one instead, some valuable navigation time could be saved.

Trap Avoidance: In my hypothesis, I did not expect to see any other pattern besides being trapped but the graphs showed a third distinct pattern. After getting trapped a few times, almost all users would try to avoid trapping by making extreme vertical (downward or upward) movements. These movements were noticed in two cases: (a) to get out of the trap and (b) to avoid getting trapped.

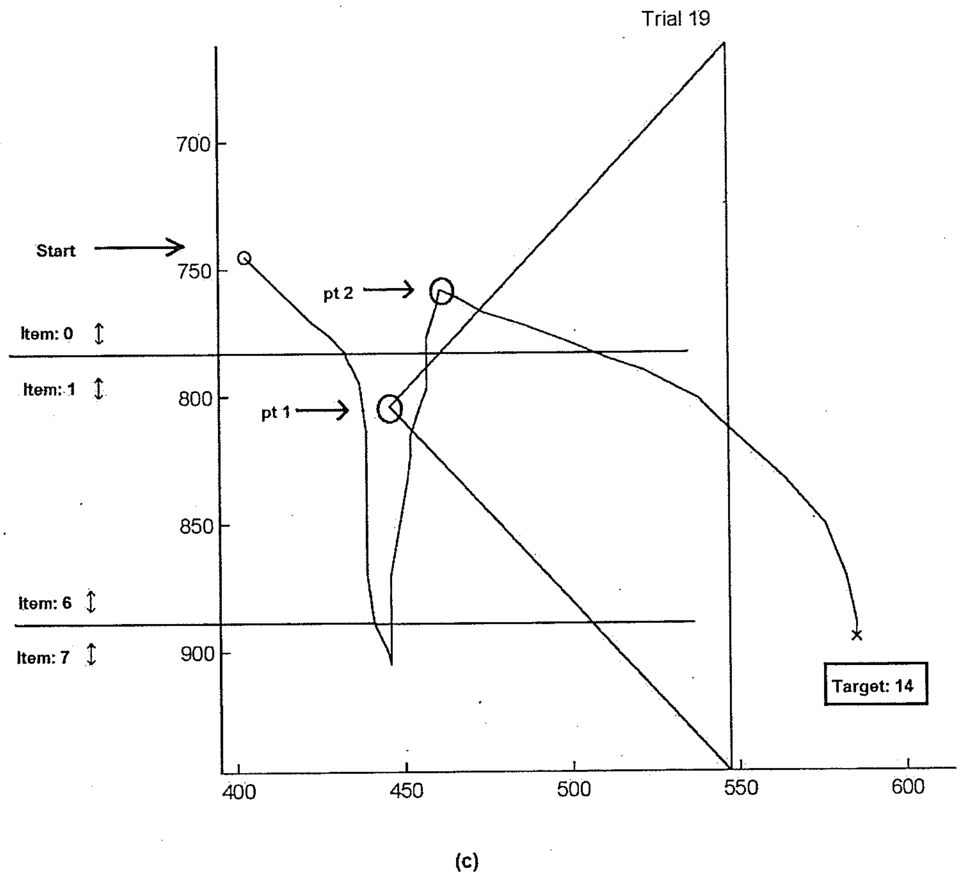

Figure 19: A graph showing trap avoidance in a trial. A quick vertical movement between item 0 and item 7 shows a user is trying to avoid getting trapped by skipping the adjacent parent items.

In figure 19, the graph shows a navigation pattern of trap avoidance even before any trapping occurred. In the trial, the two adjacent parent items were item 0 and item 1. Since item 0 led to the target item, technically the user was never trapped while moving downwards into the item but having experienced getting trapped in previous trials the user quickly leaped downwards vertically, almost skipping seven items before stopping and moving up again. On the way up, the user activated item 1 at "pt 1", and kept moving the cursor upwards until "item 0" was activated. The user then entered the newly activated AAMU and clicked on the target item. This way the user, when not ready, avoided getting trapped within the first two items and later approached the items in a different way to minimize trapping cost.

Based on the patterns of trials where users were trapped and/or tried to avoid trapping, it was clear to me that users were well aware of the trapping problem and its consequences on the interaction. With time, almost all users were able to identify menus designed to cause trapping and all users tried either: to apply some strategy to avoid trapping, or a shortcut to get out of the trap.

5.3.2 *Developing Alternate Designs*

The analysis of the results of experiment three suggested that cursor trapping is slowing down the menu navigation and selection process, with AAMUs. The user behavior clearly showed that providing an alternate path or a shortcut out of the trapping would be the best solution.

I used the following factors: shape, visual cue and AAMU drawing position (also described in the methodology section) to design the following three alternate designs:

AAMU-Click

AAMU-Click appeared identical to traditional AAMUs and it provided a shortcut path (click) to users for getting out of the trap. Previously if users found themselves trapped inside an AAMU triangle they had no choice but to backtrack or move the cursor outside the triangle and then reposition the cursor. That process not only used up some navigation time but also hindered in the interaction process of users. AAMU-Click allow the users to continue interacting with other items in the menu while staying inside the AAMU triangle. A single click on any item makes the old AAMU triangle disappear and activate the current item function, see Figure 20 . In the figure (a), a menu with two adjacent parent items, item 0 and item 1 is shown. Currently, item 1 is active but the desired item is item 0. Therefore the user, while staying inside the activation area, points the cursor to item 0 and click on it. The click action makes the activation area, associated with item 1, disappear and activates item 0 as shown in figure (b). Also, MS Windows uses mouse clicking for overriding time delay in menus, hence users can very well relate to this function.

AAMU-Hover

Another way of resolving cursor trapping would be to provide users with an alternate path using a visual cue. I designed another variant

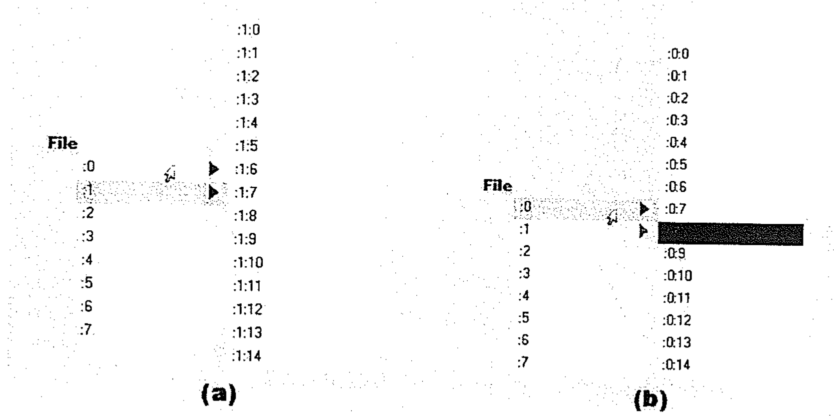

Figure 20: Example of AAMU Click (a) Before the click action: Cursor is trapped inside activation area of item 1 whereas the desired submenu is associated with item 0.
(b) after click action: While staying inside the old activation area, the user clicked on item 0 and activated it.

called AAMU-Hover. When users were trapped inside an AAMU triangle they could find a shortcut out of the trap by pointing the cursor to another parent item. As soon as the cursor crosses over the new parent item, a small arrow appeared inside the AAMU triangle. This arrow was used as a visual cue for the users to let them know that they can activate a different parent item by hovering their cursor onto the arrow, see Figure 21. In the figure the user is trapped inside the triangle of 'item 0' where as the desired item is 'item 1'. While staying inside the AAMU triangle, as soon as the cursor enters the boundaries of 'item 1' the arrow appeared. If the cursor hovered onto the arrow, the old triangle would disappear and the AAMU triangle for 'item 1' will be activated. This design also provides a shortcut to the trapping problem without interrupting the interaction process. An example of AAMU-Hover is shown in Figure 21.

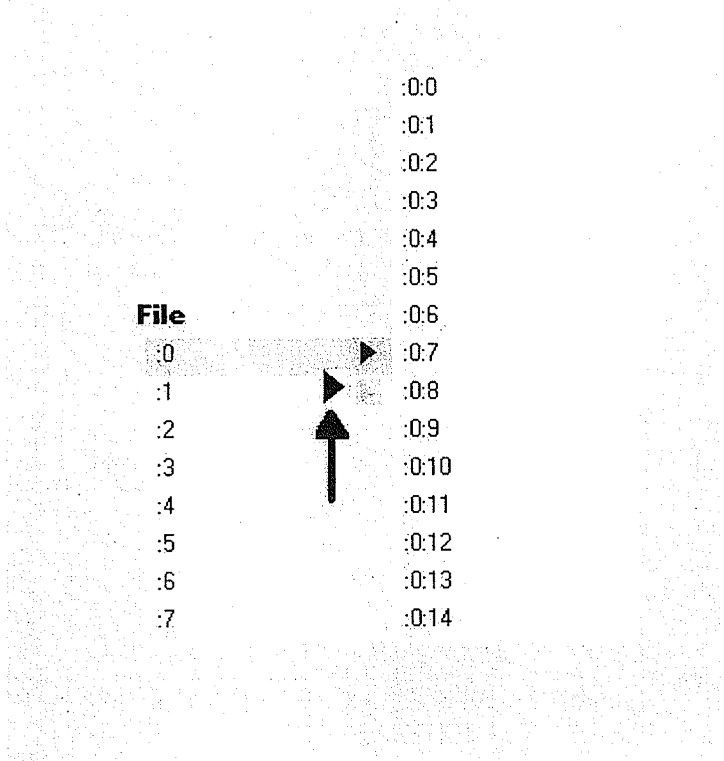

Figure 21: A two-level deep menu showing AAMU hover.

AAMU-Curve

Finally, considering the shape factor, I designed a curved version of traditional AAMUs. Instead of an equilateral triangle, the legs joining the cursor position and the top and the bottom of the child submenu were drawn as curves. The triangle was drawn a few pixels ahead of the cursor, so that user can explore the child submenu without entering the triangle or getting trapped. Even incase of trapping, the narrow tip of the curved shape makes it easier for the user to get out of the triangle serving as a quick shortcut. An example of AAMU-Curve is shown in Figure 22.

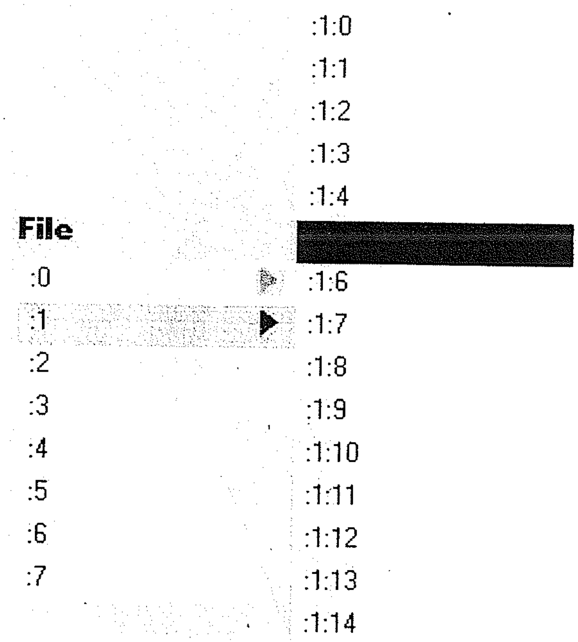

Figure 22: A two-level deep menu showing AAMU-Curve.

5.3.3 *Experiment4: Testing the Alternate Designs*

In this experiment the alternate designs discussed in section 5.3.2 were tested against traditional AAMUs to measure performance benefits among all designs. From now on, I will refer to the alternate designs as "AAMU variants". In this experiment, the menu navigation and selection tasks were designed such that all of them essentially had trapping (trapped case). The aim of this experiment was to pick the best performing AAMU variants for the final test against the default technique.

Data Recorded:

- Completion time
- Task type

- Menu type

Method: The experiment was conducted on Windows XP using a Pentium 4 machine with 1 GB of RAM. The experiment was performed using a mouse.

Participants: Twenty-five university undergraduate students participated for exchange of course credit. 13 participants performed the search task and 12 performed the selection task. All of them had prior experience using MS Windows default menu and were familiar with operating a mouse. None were color blind.

Task:

The experiment was conducted using both search and selection tasks as described in section 5.3.1.

Design of the experiment: For depth level two, there was only one possible target, whereas in level one there were two possible parent items hence, the total number of possible paths was 2. The experiment design was 4 × 2, (4 menu types, 2 task types), mixed design. All participants used all four menu types but each person only did one task type. The experiment was counterbalanced using a Latin square to eliminate the bias for menu types.

Findings of Experiment4

A total of 2160 trials were used for result analysis. Total number of trials with search task were 1200 and 960 for selection task. All data is analyzed in SPSS 16 and outliers ($-3 < range > 3$) have been removed. An ANOVA for each task type was conducted to compute significance between all menu types.

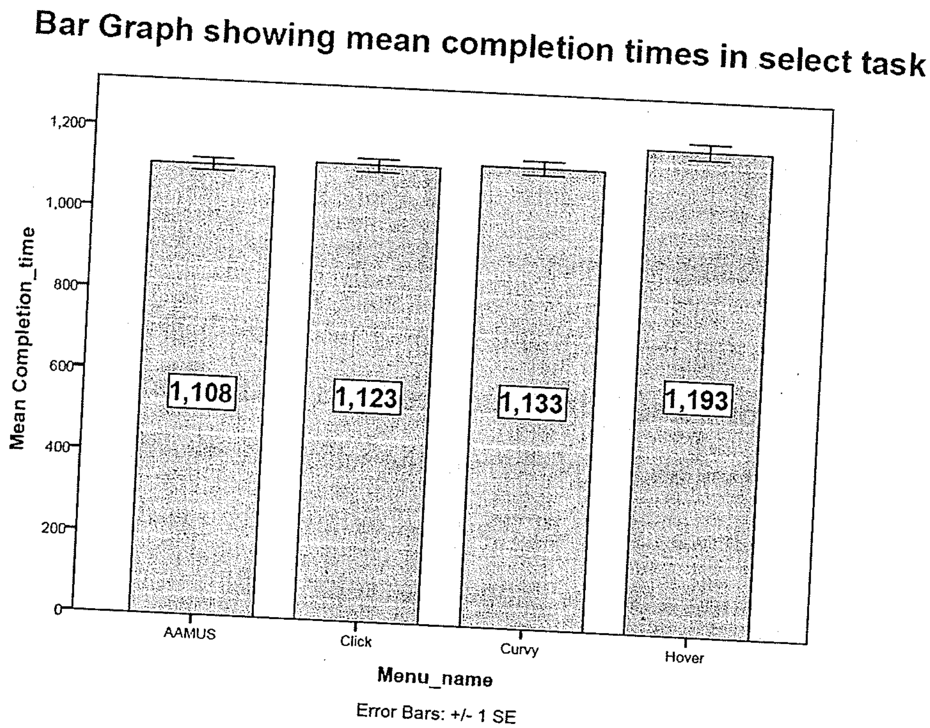

Figure 23: A bar graph showing mean completion times in select task, in experiment 4.

The results for each task type are analyzed separately as follows:

Selection Task:

In the selection task, users have the advantage of knowing the parent item that leads to the target item. Therefore, chance of getting trapped are lower in this task type,

An ANOVA for the select task, with dependent variable as "completion time", and "menu type" as independent variable, showed marginal significant main effect between menu types ($F(3,33)=2.524$, $p=.075$). However, there was a significant interaction between Menu types × Subject ($F(33,889)=2.513$, $p<.001$)(see appendix A.4 for details).

Individual menu performances were compared using pairwise comparisons, using a Bonferroni adjustment. The comparisons showed AAMU-Hover (mean 1.193s, sd 0.298) performing significantly worse than all other menus, i.e., AAMUs(p=.000), AAMU-Click(p=.007) and AAMU-Curve(p=.019). There was no significance among AAMU (1.108s, 0.242), AAMU-Click (1.123s, 0.248) and AAMU-Curve (1.133s, 0.262) (see Figure 23).

Search Task:

In the search task, users have to explore all the parent items to find the one that leads to the target item. Therefore, chance of getting trapped are higher in this task type and searching takes longer than selection.

An ANOVA for search task, with dependent variable as "completion time", and "menu type" as independent variable, showed a significant main effect between menu types ($F(3,36)=3.743$, p=.019). There was also a significant interaction between Menu types × Subject interaction ($F(36,1145)=3.161$, $p<.001$). See appendix A.4 for details.

The pairwise comparisons, using Bonferroni adjustments, showed that AAMU-Hover (1.852s, 0.981) performed significantly worse than AAMU-Curve (mean 1.580s, sd 0.672) at ($p<.001$) and AAMUs (1.637s, 0.695) at ($p<.001$) and marginally worse than AAMU-Click (1.72s, 0.845) at (p=.075). Also, AAMU-Curve performed significantly better than AAMU-Click at (p=.014). However, there was no significance among AAMU and AAMU-Curve as well as AAMU and AAMU-Click, (see Figure 24).

Conclusion:

From the above results, I concluded that AAMU-Hover was not a good design as it performed worse than traditional AAMUs. A major problem with the hover technique was that the arrow would appear on all diagonal movements, even when users were not actually

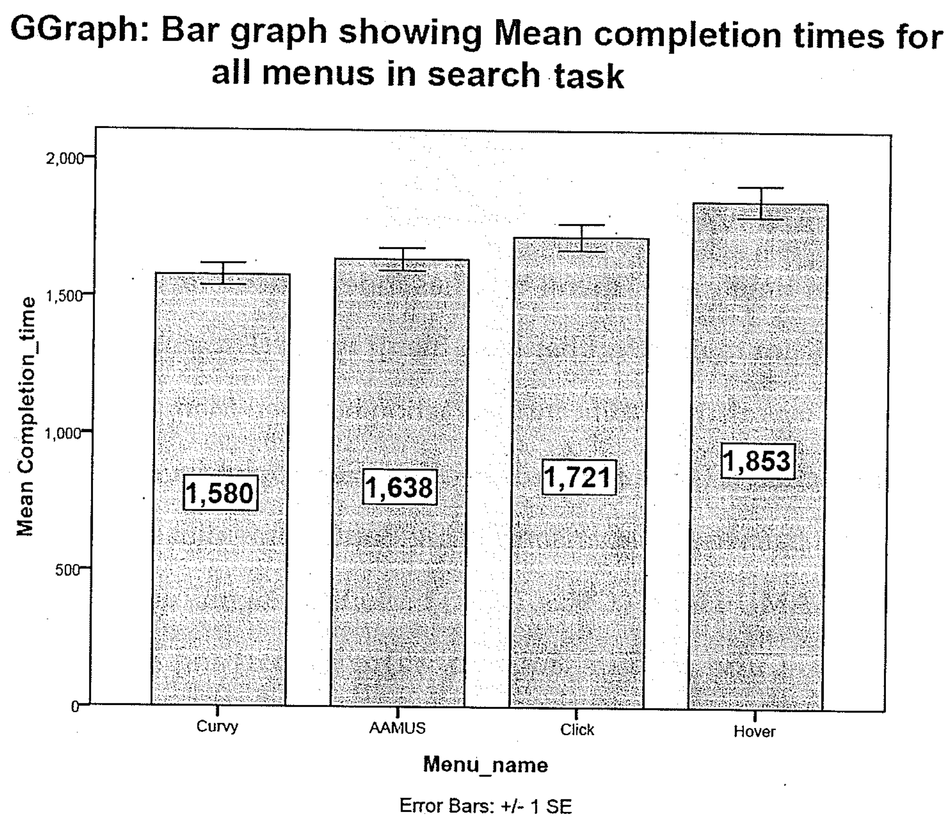

Figure 24: A bar graph showing mean completion times in the search task, in experiment 4.

trapped. This way, the arrow interfered with users' diagonal motion and it caused more unintended submenu invocations/ revocations when the users accidentally hovered on the arrow while navigating. These unexpected AAMU activations contributed to higher completion times. Therefore, I decided to not include AAMU-Hover in the final test.

Among AAMU-Curve and AAMU-Click, both showed certain benefits in both selection and searching. I found that the curve shape helped in search task whereas the click helped in selection and I decided to combine the curve shape and the click function to get the most benefit of the two designs. The new design will be called AAMU-Curve-Click.

5.3.4 *Experiment5: Putting the Best Designs to the Final Test*

A controlled experiment was conducted to evaluate the benefits of AAMUs and its 2 variants, AAMU-Curve and AAMU-Curve-Click, against the default technique in cascading pull-down menus.

Method:

The experiment was conducted on Pentium 4 desktop computers running Windows XP operating system. A full screen-color mode with a 1024 × 768 resolution was used. Two input devices were used: a conventional optical mouse and a touch pad. All default system settings for the three devices were used.

Participants: 52 undergraduate students participated in the experiment for exchange of course credit. 20 of them performed on the touch pad and 32 performed using the mouse. All participants were experienced computer users, using mouse on a daily basis. None were color blind.

Task:

- The experiment was conducted using both search and selection tasks.
- In the search task, users were required to browse all the parent items (in other words, all possible paths) until the target item was located and a single click on the target item completed the task. No visual cues are provided for the path. The target item in both tasks was highlighted in red. This was similar to that of experiment 2.
- In the selection task, the path to be followed was highlighted in green and the user would follow the path until the target item is located. This was similar to that of experiment 1.
- Each trial started when the user clicked on the "File" button in the center of the screen. The menu was displayed upon click, users navigate inside the menu until the target item was located and a single click on the target item completed the task. The trial would not end unless the target item was clicked. All other clicks were recorded as error.
- Each task was timed. Timer started when the user clicked on "File" button and ended with the click on target.
- Each menu in every task was two levels deep. In the first level there were 10 items and fifteen items in the second level.

- In the trapped case, there were two adjacent parent items located to ensure trapping. However, in the clear case, only one parent item within a menu was placed so that no trapping can occur.

- The menus were drawn in the center of the screen to enable center alignment in all scenarios.

An example of a trial in this experiment is shown in Figure 25.

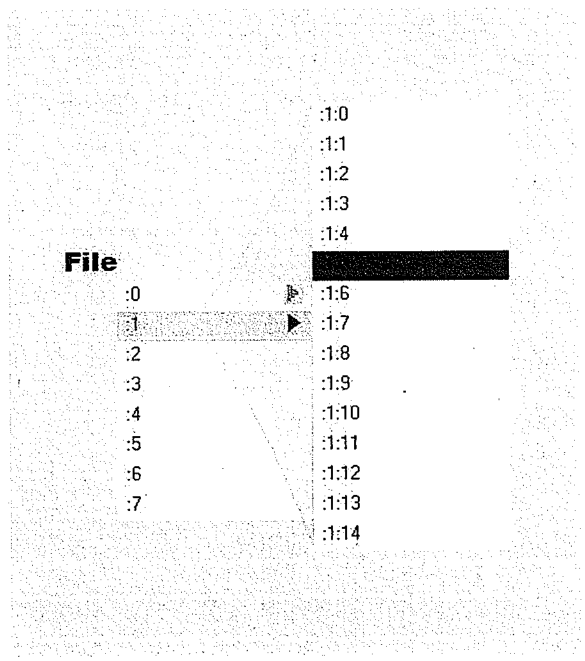

Figure 25: An example of a task in the final experiment.
A click on the "File" button activated the menu and a click on the target highlighted in red ended the trial.

In order to test all designs on fair grounds, all trials have the target item at the eighth position. The parent positions varied between positions three, five and seven.

Design of the experiment:

Application Names	Clear/Trapped	Number of Menus						Total	% Ratio
		File	Edit	View	Insert	Format	Tools		
Microsoft Word	Clear	2	1	1	3	2	1	10	90.909
	Trapped	0	0	0	0	0	1	1	9.091
	Total	2	1	1	3	2	2	11	**91-9**
Microsoft Excel	Clear	3	0	1	2	0	2	8	66.667
	Trapped	0	1	0	0	1	2	4	33.333
	Total	3	1	1	2	1	4	12	**67-32**
Microsoft Power Point	Clear	2	0	1	2	1	0	6	75.000
	Trapped	0	0	1	0	0	1	2	25.000
	Total	2	0	2	2	1	1	8	**75-25**
Microsoft Outlook	Clear	1	0	2	0	0	1	4	66.667
	Trapped	1	0	0	0	0	1	2	33.333
	Total	2	0	2	0	0	2	6	**67-32**
Microsoft Internet Explorer	Clear	2	0	1	0	0	1	4	57.143
	Trapped	0	0	2	0	0	1	3	42.857
	Total	2	0	3	0	0	2	7	**57-42**
Mozilla FireFox	Clear	0	0	2	0	0	1	3	75.000
	Trapped	0	0	1	0	0	0	1	25.000
	Total	0	0	3	0	0	1	4	**75-25**
Total	**Clear**							35	72.9166667
	Trapped							13	27.0833333
							Total	48	**73-27**

Figure 26: A comparison of clear versus trapped paths in various commonly used applications.

The objective of this study was to compare all techniques in a real world scenario therefore, I conducted a survey of most commonly used applications and calculated an average ratio of clear versus trapped paths (see Figure 26).

The applications surveyed included MS Word, MS Excel, MS Power Point, MS Internet Explorer and Mozilla Firefox. Six commonly used menus were analyzed namely file, edit, view, insert, format and tools. Out of total 48 paths, 35 were clear and only 13 were trapped. The average ratio for clear:trapped was found to be 73 : 27. When looking individually, out of six applications, only one has clear:trapped ratio below 60 : 40. Three applications have the ratio for clear:trapped either equal or higher than 75 : 25 and for the rest of the three applications the average ratio was 60 : 40. Hence for this experiment, I tested both ratios among all techniques.

Each task, select and search, was divided into two blocks: "block 1" with ratios(75:25) and "block 2" with ratios(60:40) whereas, 60 and 75 describes the percentage of clear trials and 40 and 25 describes percentage of trapped trials in the block. Each participant performed two test sessions, one per each task and using one device. The order of testing tasks and blocks was counterbalanced between all 53 participants. Participants were allowed to take rests between trials and test sessions. Before the test began, all participants were allowed to have as many practice trials as they needed to get used to the device and to gain sufficient practice skill.

A test session consisted of 264 trials which were divided into two blocks where as block 1 included 104 actual trials and 10 practise trials and block 2 included 140 actual and 10 practise trials. Within each block, all trials were performed four times, one for each menu type. The order of menus was randomized using a Latin-square. Each session lasted about 30 minutes. After each block was completed, a recess screen was shown, and the participant could take rest if desired.

The total number of trials in the experiment can be computed as follows:

53 participants $\times$ 2 tasks $\times$ 2 blocks $\times$ 4 menus types

where as total trials in both blocks were 264 so the result can be computed as:

$53 \times 2 \times 264 \times 4 = 111936$

and out of these 8480 were practice trials and excluded from result analysis.

The total selection time for each trial was measured in milliseconds.

Hypothesis: Completion times for 60:40 ratio should be higher than 75:25 in all menu types and that either one or both AAMU variants should outperform AAMUs and default technique in completion time for both ratios.

Findings of Experiment5:

All data is analyzed in SPSS 16 and outliers ($-3 < range > 3$) have been removed. AAMUs and its variants showed lowest completion times in all devices and tasks. Over all mouse (mean 1.41s, sd 0.519) among devices and selection (mean 1.601s, sd 0.672) among tasks had lowest completion times in all menu types. The mean completion times with respect to Menu types, device types and task types are available in appendix A.5.

An overall ANOVA (for each device separately) ,with dependent variable as "completion time" showed a significant main effect between menu types, i.e.,for mouse:($F(3,93)=116.83$, $p<.001$) and for touch pad: ($F(3,57)=63.839$, $p<.001$). There was also a significant main effect for task types, i.e, for mouse ($F(1,31)=242.581$, $p<.001$) and for touch pad: ($F(1,19)=282.618$, $p<.001$). However there was no significance found between Menu$\times$Task types interaction for mouse ($F(3,93)=0.77$), $p=.513$) but significance for touch pad: ($F(3,57)=2.726$, $p=.052$).

5.3.5 *Evaluation of the Data:*

I then analyzed the rest of the data for each device separately. All participants performed two tasks with each device, namely item selection and item searching. Each task was performed with two cascading ratio settings.

I will first describe the results with respect to device and then task type and ratio respectively.

Mouse:

In the mouse device, overall means are shown in Figure 27).

A pairwise comparison, using Bonferroni adjustments, showed all AAMU variants perform significantly faster than default technique at (p<.001). Also, AAMU-Curve-Click was significantly better than AAMU-Curve and AAMU at (p=.015 and p=.038) respectively. Post-hoc comparisons are showed in Figure 28.

Selection:

An ANOVA for mouse device and selection task, with dependent variable as "completion time", and "menu type" and "Ratio" as independent variable, showed a significant main effect between menu types ($F(3,93)=82.592$, $p<.001$) and between the two ratios ($F(1,31)=31.236$, $p<.001$). However there was no significance found between Menu types$\times$Ratio interaction ($F(3,93)=1.275$, $p=.288$)(see appendix A.5 for details).

The results for each ratio are described as follows.

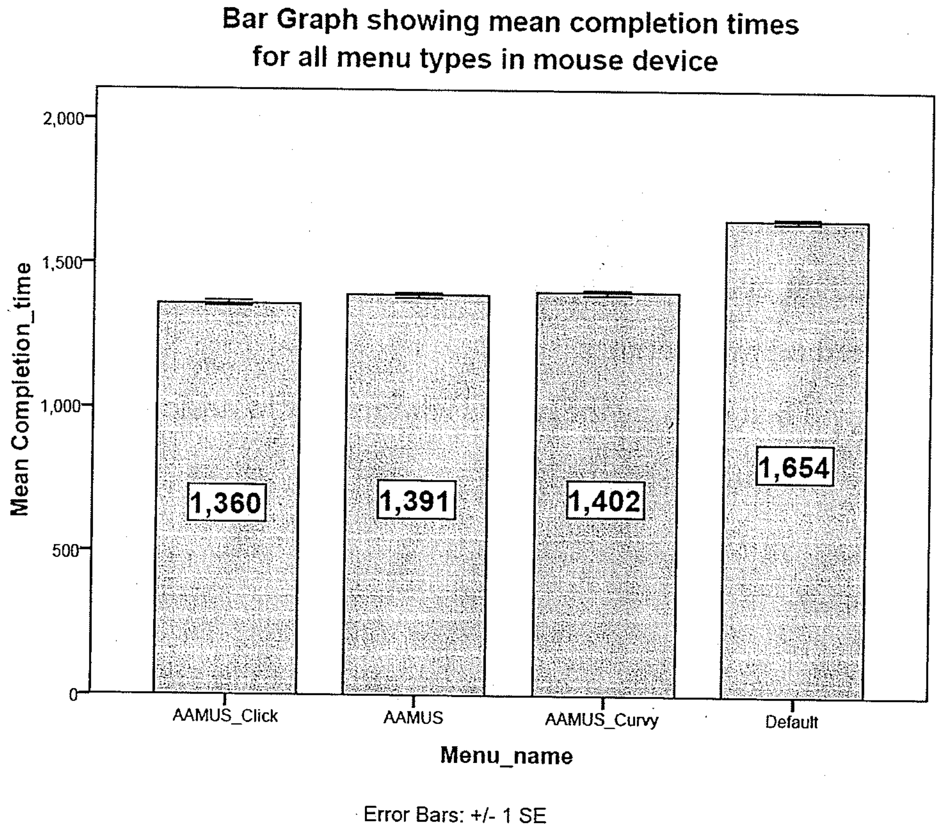

Figure 27: A bar graph representing mean completion times for all menus using mouse device.

Ratio (60:40): This is the ratio setting where possibility of getting trapped is very high. 40 percent of total tasks had adjacent parent items causing trapping. Overall mean completion time for this ratio was (mean 1.297s, sd 0.450). There was a significant main effect found for menu types ($F(3,93)=59.328$, $p<.001$). In case of individual menu performance, a pairwise comparison using Bonferroni adjustments showed default (1.525s, 0.50) performing significantly worse than AAMU-Curve-Click (mean 1.216s, sd 0.389), AAMU-Curve (1.236s, 0.416) and AAMUs (1.245s, 0.419). In this case AAMU was slightly worse than

Pairwise Comparisons

Dependent Variable:Completion time

(I) Menu name	(J) Menu name	Mean Difference (I-J)	Std. Error	Sig.[a]	95% Confidence Interval for Difference[a] Lower Bound	Upper Bound
AAMUS	AAMUS_Click	33.760*	12.366	.038	1.129	66.392
	AAMUS_Curvy	-3.606	12.358	1.000	-36.214	29.003
	Default	-263.299*	12.385	.000	-295.980	-230.618
AAMUS_Click	AAMUS	-33.760*	12.366	.038	-66.392	-1.129
	AAMUS_Curvy	-37.366*	12.354	.015	-69.965	-4.767
	Default	-297.059*	12.381	.000	-329.730	-264.389
AAMUS_Curvy	AAMUS	3.606	12.358	1.000	-29.003	36.214
	AAMUS_Click	37.366*	12.354	.015	4.767	69.965
	Default	-259.693*	12.372	.000	-292.341	-227.046
Default	AAMUS	263.299*	12.385	.000	230.618	295.980
	AAMUS_Click	297.059*	12.381	.000	264.389	329.730
	AAMUS_Curvy	259.693*	12.372	.000	227.046	292.341

Based on estimated marginal means

*. The mean difference is significant at the .05 level.

a. Adjustment for multiple comparisons: Bonferroni.

Figure 28: A pairwise comparison showing significance between different menu types for mouse device.

its variants however, there was no significance found among AAMUs and its variants.

Ratio (75:25): In this ratio setting only 25 percent of total tasks had adjacent parent items and hence low possibility of trapping. The results supported the hypothesis that overall mean completion time for this ratio (mean 1.235s, sd 0.421) would be less than the mean completion time of 60:40 ratio. There was a significant main effect found for menu types ($F(3,93)=37.41$, $p<.001$) on completion time. In case of individual menu performance, a pairwise comparison using Bonferroni adjustments showed default (1.439s, 0.489) performing significantly worse than AAMUs (mean 1.156s, sd 0.319), AAMU-Curve-Click (1.166s, 0.383) and AAMU-Curve (1.208s, 0.489). However, there was no significance found among AAMU and its variants. In this case

AAMU was slightly better than its variants in case of little or no trapping.

Between the two ratios, only AAMUs and AAMU-Curve-Click showed significant improvement in mean completion times. The graph for mean completion times in both ratios and all menu types are shown in Figure 29.

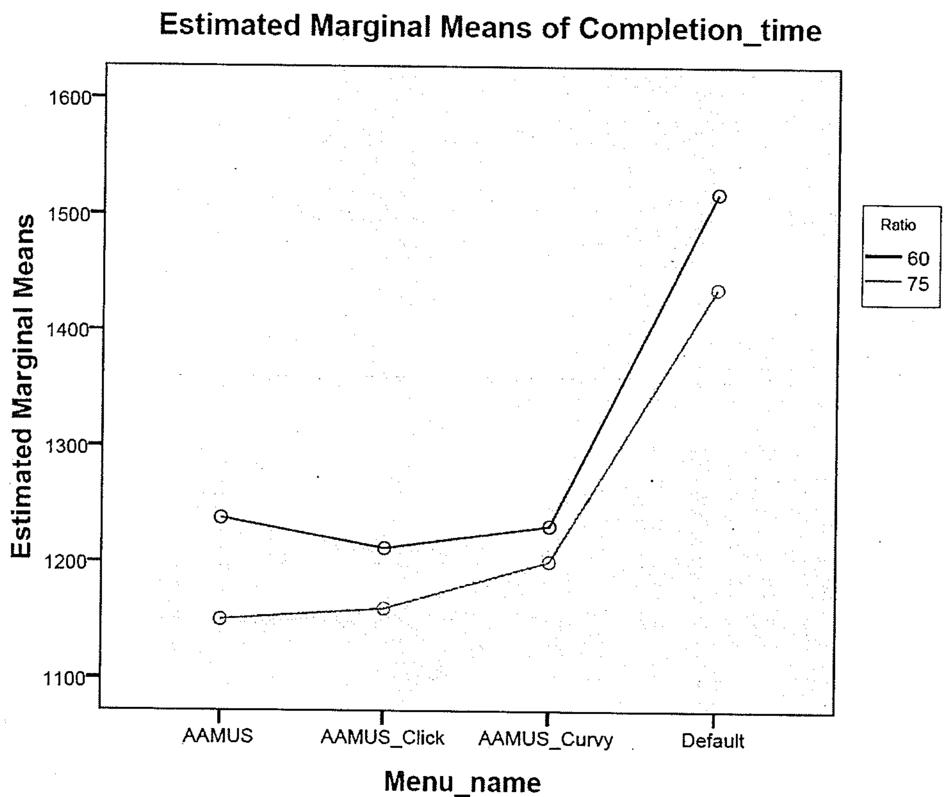

Figure 29: A graph showing estimated mean completion times for all menu types in both ratios, in case of mouse device and selection task.

Searching:

An ANOVA for mouse device and search task, with dependent variable as "completion time", and "menu type" and "Ratio" as independent variable, showed a significant main effect between menu types ($F(3,93)=58.482$, $p<.001$) and between the two ratios ($F(1,31)=9.674$, $p=.004$). However there was no significant interaction effect between Menu types×Ratio ($F(3,93)=0.534$, $p=.660$). See appendix A.5 for details.

Ratio (60:40): Mean completion time for this ratio was (mean 1.657s, sd 0.539). There was a significant main effect found for menu types ($F(3,93)=31.63$, $p<.001$). In case of individual menu performance, a pairwise comparison using Bonferroni adjustments showed default (1.867s, 0.527) performing significantly worse than AAMU-Curve-Click (mean 1.570s, sd 0.522), AAMUs (1.619s, 0.547), and AAMU-Curve(1.636s, 0.522). AAMU-Curve-Click was also significantly better than AAMU-Curve at ($p=.054$). The result shows that when it comes to searching with trapped cases, AAMUs-Curve-Click have a slightly better performance than traditional AAMU and its curved variant. One reason might be that in searching, there are more chances of trapping and the click version provided the benefit of shortcut for getting out of the trap and hence showed the better completion times.

Ratio (75:25): Mean completion time for this ratio was (mean 1.580s, sd 0.514). There was a significant main effect found for menu types ($F(3,93)=31.528$, $p<.001$). In case of individual menu performance, a pairwise comparison using Bonferroni adjustments showed default (1.789s, 0.565) performing significantly worse than AAMU-Curve-Click (mean 1.486s, sd 0.453), AAMU-Curve (1.506s, 0.446) and AAMUs (1.548s, 0.543). There was no significance found between AAMU and its variants.

Between the two ratios, all menu types showed significant improvement in mean completion times. The mean completion times for both ratios and all menu types are shown in Figure 30.

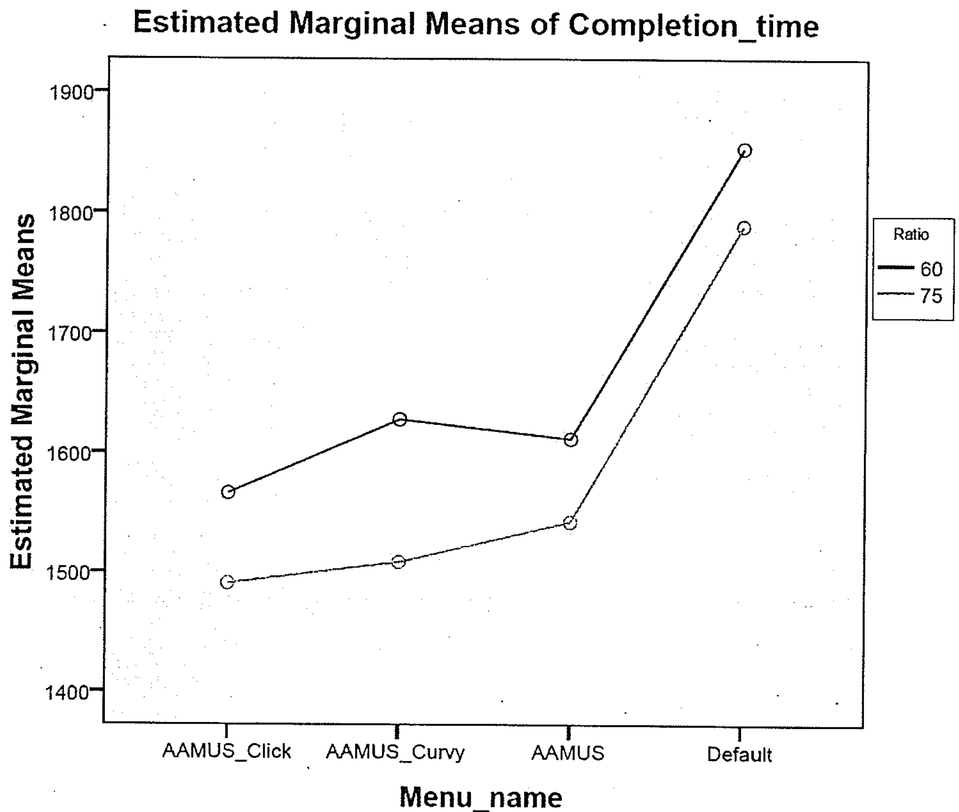

Figure 30: A graph showing estimated mean completion times for all menu types in both ratios, in case of mouse device and search task.

Touch Pad:

In the touch pad device, overall means are shown in Figure 31).

A pairwise comparison, with Bonferroni adjustments, showed showed all AAMU variants perform significantly faster than default technique at (p<.001). Although AAMU-Curve-Click showed lowest

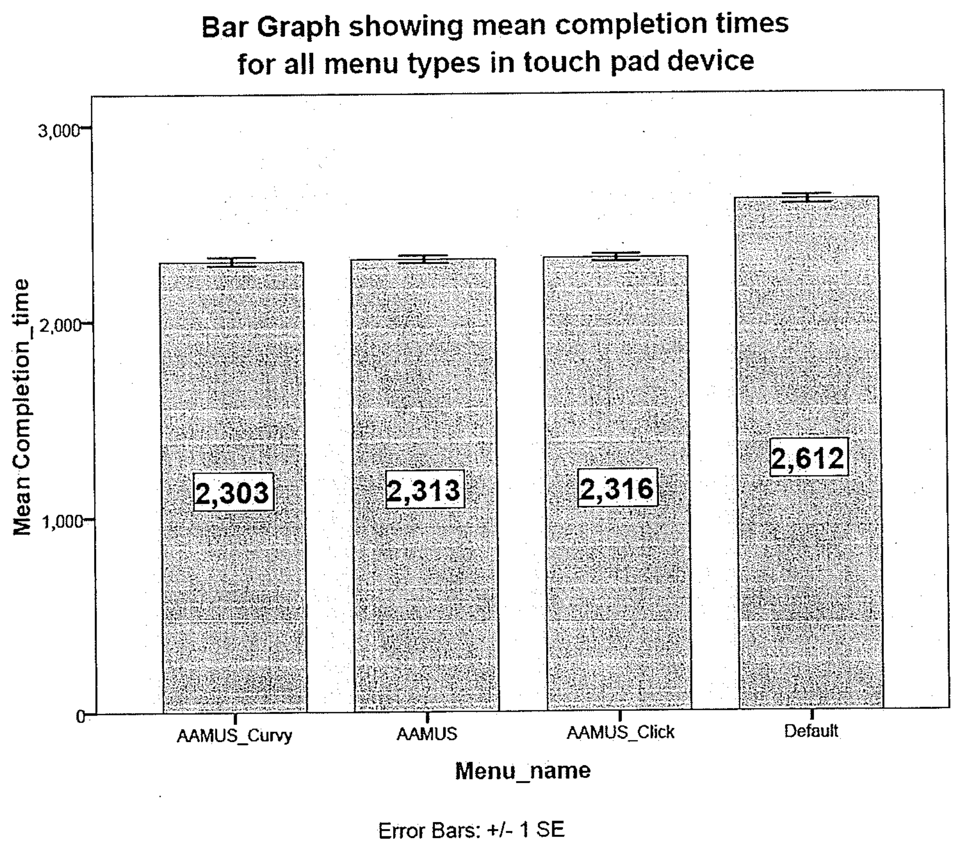

Figure 31: A bar graph representing mean completion times for touch pad device.

completion times among all menu types but there no significance among AAMU and its variants (see Figure 32).

Selection:

An ANOVA for touch pad device and selection task, with dependent variable "completion time", and "menu type" and "Ratio" as independent variable, showed a significant main effect between menu types ($F(3,57)=26.97$, $p<.001$). However there was no significance found between the two ratios ($F(1,19)=2.17$, $p=.157$) and Menu types$\times$Ratio interaction ($F(3,57)=.333$, $p=.801$)(see appendix A.5 for details).

Pairwise Comparisons

Dependent Variable:Completion time

(I) Menu name	(J) Menu name	Mean Difference (I-J)	Std. Error	Sig.[a]	95% Confidence Interval for Difference[a] Lower Bound	Upper Bound
AAMUS	AAMUS_Click	3.543	22.438	1.000	-55.674	62.760
	AAMUS_Curvy	1.966	22.454	1.000	-57.292	61.224
	Default	-306.241*	22.542	.000	-365.730	-246.751
AAMUS_Click	AAMUS	-3.543	22.438	1.000	-62.760	55.674
	AAMUS_Curvy	-1.577	22.401	1.000	-60.695	57.541
	Default	-309.784*	22.488	.000	-369.133	-250.435
AAMUS_Curvy	AAMUS	-1.966	22.454	1.000	-61.224	57.292
	AAMUS_Click	1.577	22.401	1.000	-57.541	60.695
	Default	-308.207*	22.504	.000	-367.597	-248.816
Default	AAMUS	306.241*	22.542	.000	246.751	365.730
	AAMUS_Click	309.784*	22.488	.000	250.435	369.133
	AAMUS_Curvy	308.207*	22.504	.000	248.816	367.597

Based on estimated marginal means

a. Adjustment for multiple comparisons: Bonferroni.

*. The mean difference is significant at the .05 level.

Figure 32: A pairwise comparison showing significance between different menu types for touch pad device.

Ratio (60:40): Mean completion time for this ratio was (mean 2.17s, sd 0.643). There was a significant main effect found for menu types ($F(3,57)=15.822$, $p<.001$). In case of individual menu performance, a pairwise comparison using Bonferroni adjustments showed default (2.368s, 0.7) performing significantly worse than AAMU-Curve performed (mean 2.077s, sd 0.602), AAMUs (2.081s, 0.580) and AAMU-Curve-Click(2.156s, 0.642). However there was no significance found among AAMU and its variants The results clearly show that in touch pad device, click action interrupts and slows down the interaction process, hence when there are more trapped tasks, AAMU-Curve-Click is the slowest to complete among its variants.

Ratio (75:25): Mean completion time for this ratio was (mean 2.112s, sd 0.626). There was a significant main effect found for menu types ($F(3,57)$=16.275, p<.001). In case of individual menu performance, a pairwise comparison using Bonferroni adjustments showed default (2.344s, 0.692) performing significantly worse than AAMUs (mean 2.013s, sd 0.556), AAMU-Curve (2.026s, 0.571), and AAMU-Curve-Click(2.062s, 0.610) . However there was no significant difference among AAMU and its variants. Therefore, it can be concluded that when there are not much trapped cases, in touch pad device, then all three AAMU versions perform equally.

Between the two ratios, no menu types showed significant improvement in mean completion times. The mean completion times for both ratios and all menu types are shown in Figure 33.

Searching:

An ANOVA for touch pad device and selection task, with dependent variable "completion time", and "menu type" and "Ratio" as independent variable, showed a significant main effect between menu types ($F(3,57)$=44.272, p<.001). Also, there was significance found between the two ratios ($F(1,19)$=10.588, p=.004) but no significance among Menu types×Ratio interaction ($F(3,57)$=1.105, p=.355)(see appendix A.5 for details).

Ratio (60:40): Mean completion time for this ratio was (mean 2.665s, sd 0.735). There was a significant main effect found for menu types ($F(3,57)$=15.11, p<.001). In case of individual menu performance, a pairwise comparison using Bonferroni adjustments showed default (mean 2.890s, sd 0.785) performing significantly worse than AAMU-Curve-Click performed fastest (mean 2.544s,

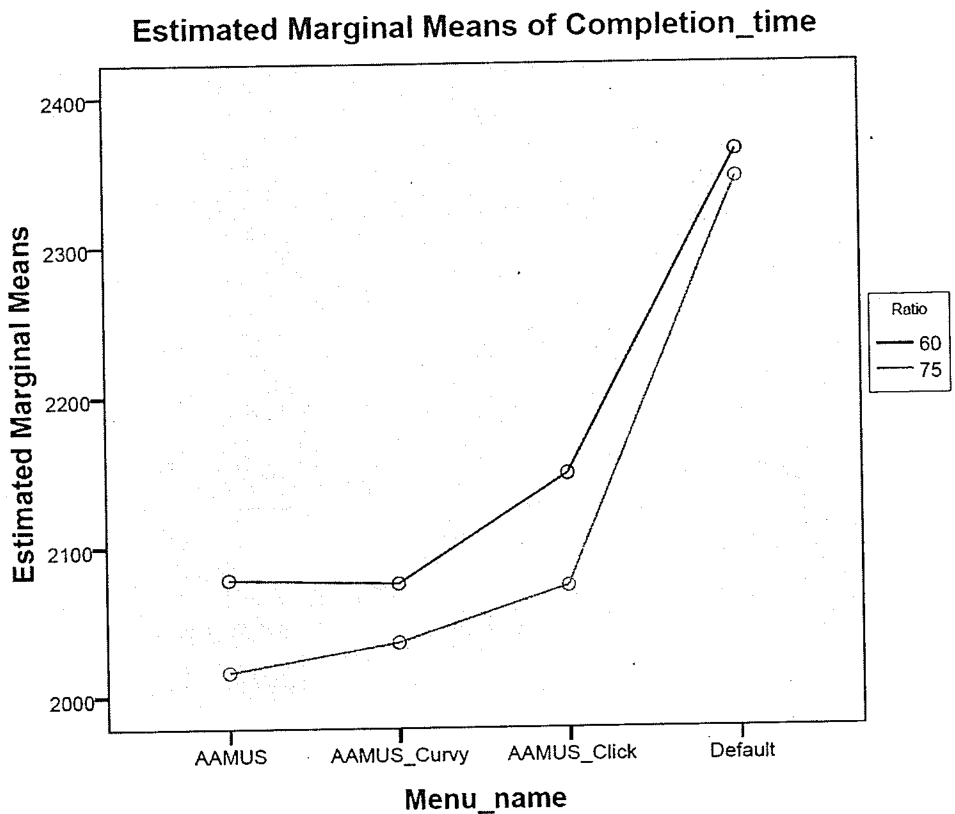

Figure 33: A graph showing estimated mean completion times for all menu types in both ratios, in case of touch pad device and select task.

sd 0.681), AAMUs (2.604s, 0.691), and AAMU-Curve(2.642s, 0.731). There was no significant difference among AAMU and its variants. The results are consistent with mouse searching. Since in searching the users are only exploring the various child menus while staying inside the parent menu, AAMU-Curve-Click provided the quickest way out of the activation area as compare to the other two designs.

Ratio (75:25): Mean completion time for this ratio was (mean 2.564s, sd 0.712). There was a significant main effect found for menu types ($F(3,57)=24.880$, $p<.001$). In case of individual menu performance, a pairwise comparison using Bonferroni adjustments showed default (2.847s, 0.759) performing significantly worse than AAMU-Curve performed fastest (mean 2.444s, sd 0.671), AAMU-Curve-Click (2.446s, 0.671) and AAMU(2.521s, 0.668). However, there was no significant effect among AAMU and its variants. AAMUs performed slightly worse than its variants which can be attributed to the shape of AAMU triangle. Therefore, it can be concluded that when there are not much trapped cases, in touch pad device, then all three AAMU versions perform equally.

Between the two ratios, all menu types, except default, showed significant improvement in mean completion times. The mean completion times for both ratios and all menu types are shown in Figure 34.

5.3.6 *Conclusion:*

Overall in all device and task groups, AAMU and its variants performed significantly better than the default technique. The results also supported both of the following hypothesis:

- Completion times for 60:40 ratio should be higher than 75:25 in all menu types.
- Either one or both AAMU variants should outperform AAMUs and default for 60:40 ratio.

Although there was no significance among AAMU and its variants but there was improvement in performance in case of trapped trials. The individual menu performance across both devices and task types remained consistent as well. On the mouse and with both tasks,

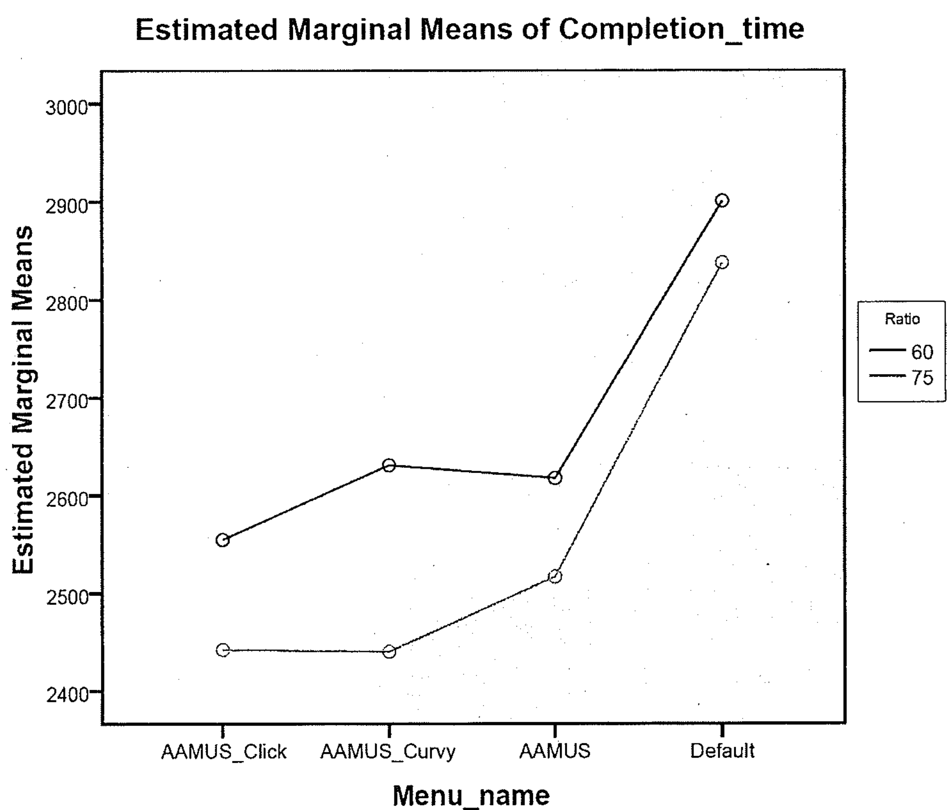

Figure 34: A graph showing estimated mean completion times for all menu types in both ratios, in case of touch pad device and search task.

AAMU-Curve-Click performed better showing the advantage of combining the curved shape and "click" shortcut in case of trapping. Whereas, on touch pad the "click" cost extra time due to different device type. Therefore, while in the search task, AAMU-Curve-Click remained the fastest, in selection task AAMUs and AAMU-Curve took over.

The results also supported my decision of combining the curve shape and click function, as in experiment 2 5.3.3, AAMU-Curve showed lowest completion times in the search tasks but in this experiment the combination design showed even lower times the AAMU-Curve alone.

The overall menu performances for both devices and task types are summarized in table 1. Four stars indicate the best performing menu type in a particular device, task and cascading density combination. Three stars indicate the second best and so on.

		Mouse		Touch Pad	
		Select	Search	Select	Search
High Density (60:40)	AAMU-Curve-Click	****	****	**	****
	AAMU-Curve	***	**	****	**
	AAMU	**	***	***	***
	Default	*	*	*	*
Low Density (75:25)	AAMU	****	**	****	**
	AAMU-Curve-Click	***	****	**	***
	AAMU-Curve	**	***	***	****
	Default	*	*	*	*

Table 1: Summary of the performance of all menu types in both mouse and touch pad device, search and selection task and high and low cascading densities.

6

SUMMARY AND FUTURE WORK

6.1 SUMMARY

In this research, I created a new technique, I called AAMUs or adaptive activation area menus. AAMUs are aimed at improving the performance of cascading menus in graphical user interfaces (GUIs). The AAMU design introduced a triangle shaped "adaptive activation area" to eliminate corner steering in traditional cascading menus. I also designed a variant of AAMUs with force-fields called force-AAMU. As described in chapter 4, two experiments were conducted to measure performance benefits of AAMU and force-AAMU against traditional cascading menus also known as the default technique and other existing techniques, namely enlarged activation area menus or EMUs [6], gesture-based menus [10] and force-fields [3]. AAMUs showed significant improvement over EMUs, default and gesture based and lower completions times than all other techniques, in users' performance in a selection task. However, I also discovered that the AAMUs suffer from a "cursor trapping" problem. When the adaptive activation area is fully expanded, the user cannot invoke the item adjacent to the currently activated item. The only option is to get out of the triangle before activating the required item. This problem slowed down the overall navigation process. To confirm that cursor trapping was the real cause for slow navigation in AAMUs I conducted another user study, described as experiment 3 in chapter 5. Based on the results of this observational study, I designed three AAMU variants, to solve cursor trapping. I named the new designs as AAMU-Click, AAMU-Hover and AAMU-Curve. A user study, described as experiment 4 in chapter 5, was conducted to test the effectiveness of the three improved designs. Based on the results, the weakest design, AAMU-Hover was eliminated and AAMU-Click and AAMU-Curve were combined into a new design AAMU-Curve-Click to get the maximum benefit of both designs. A final study, described as experiment 5 in chapter 5, was then conducted to compare performance of AAMUs, AAMU-Curve-Click, AAMU-Curve and the default technique. The final study was done using two input devices (a mouse and a touch pad), two tasks (a selection task and a searching task) and two cascading density levels (high and low). In all devices and all tasks, AAMUs and AAMU variants performed significantly better than the default technique. In high cascading density cases where more cursor trapping occurred, AAMU variants showed improved performance over AAMUs whereas in low density there was almost no difference among AAMUs and AAMU variants. Therefore, it can be concluded that the new AAMU designs helped with the cursor trapping problem.

6.2 CONTRIBUTIONS

The major contribution of this research work is the introduction of a novel menu design, AAMUs. The primary results of my research are promising and suggest that AAMUs are worth considering in GUI applications for desktop or web interfaces, and possibly small screen applications. I believe that AAMUs can also improve interactions with devices like pen or stylus.

6.3 LIMITATIONS

The benefits of every design can also be shadowed by certain limitations. Following are some problems with AAMUs that I am aware of:

- AAMUs introduced a new design in traditional cascading menus. The users have to adapt to steering in a diagonal path instead of steering through narrow tunnels and sharp corners. The observation study showed that some practise is required but generally users learn very quickly.
- A major drawback of AAMUs is the trapping problem. The AAMU triangle that is meant to provide a wider steering path, also interferes with users' interaction and causes longer navigation times. As the cascading density in a menu increases, the trapping problem occurs more frequently. Although the new AAMU designs improved the trapping problem, this is still a limitation over the traditional cascading menus.
- Finally, another limitation is the static behavior of the AAMUs. The AAMU triangle only adapts its size and position, according to the cursors' initial position, at the time of rendering. Once the cursor is inside the triangle, the triangle remains static. If

the triangle adapts its shape and size according to the users' navigation pattern, the trapping problem may be eliminated completely.

6.4 FUTURE WORK

The next step in this research direction is to develop a model of menu interaction that predicts the users' movement patterns . Based on that model more efficient and optimized variants of AAMUs can be designed. An adaptive and/or adaptable version will be the ultimate goal. Also, AAMUs can be incorporated into various interaction techniques across different applications, platforms and devices.

6.5 FINAL WORDS

Menu selection and navigation are such often performed tasks that even a small improvement in performance of these tasks will have a major effect on how users interact with current GUIs. AAMUs is a simple and easy to implement technique and is aimed at facilitating users in submenu selection, navigation and reducing the number of movement errors. This is the only technique that addresses both the problems of long and narrow steering paths as well as corner steering. Users have unanimously preferred AAMUs over the existing techniques for their simple design, ease of use and accuracy. AAMUs allow users to navigate better and with ease without introducing confusing and challenging interfaces. I believe that techniques like AAMUs can take menu navigation and selection to a whole new level. I also hope that the "keep it simple" concept of AAMUs will introduce a new paradigm in designing menus. Finally, I hope that designers will consider AAMUs as a potential way to make submenu selections easier for their users.

A

MATERIAL FROM EXPERIMENTS

A.1 EXPERIMENT1: SELECTION TASK

Which Technique did you like the most? Rate from 1 to 5, 1 being the best

- AAMUS ____
- Force AAMUS ____
- Gesture Based ____
- EMUS ____
- Default ____

What did you like the most about these Techniques? And WHY

What did you like the least about these Techniques? And WHY

What Technique do you think is the fastest? And WHY

Any Suggestions?

Post study questionnaire for Experiment 1 (Selection task with mouse).

General Linear Model

Within-Subjects Factors

Measure:MEASURE_1

Menu	Depth	Dependent Variable
1	1	AAMUS.2.00
	2	AAMUS.3.00
	3	AAMUS.4.00
2	1	Default.2.00
	2	Default.3.00
	3	Default.4.00
3	1	Direction.2.00
	2	Direction.3.00
	3	Direction.4.00
4	1	EmUs.2.00
	2	EmUs.3.00
	3	EmUs.4.00
5	1	Force.2.00
	2	Force.3.00
	3	Force.4.00
6	1	ForceAAMU.2.00
	2	ForceAAMU.3.00
	3	ForceAAMU.4.00

Multivariate Tests[b]

Effect		Value	F	Hypothesis df	Error df	Sig.
Menu	Pillai's Trace	.886	9.349[a]	5.000	6.000	.008
	Wilks' Lambda	.114	9.349[a]	5.000	6.000	.008
	Hotelling's Trace	7.791	9.349[a]	5.000	6.000	.008
	Roy's Largest Root	7.791	9.349[a]	5.000	6.000	.008
Depth	Pillai's Trace	.954	93.682[a]	2.000	9.000	.000
	Wilks' Lambda	.046	93.682[a]	2.000	9.000	.000
	Hotelling's Trace	20.818	93.682[a]	2.000	9.000	.000

a. Exact statistic

b. Design: Intercept
Within Subjects Design: Menu + Depth + Menu * Depth

Page 1

Raw statistics from SPSS for Experiment 1 (Selection task with mouse).
Summary of Repeated measures Analysis for completion times.

Multivariate Tests[b]

Effect		Value	F	Hypothesis df	Error df	Sig.
Depth	Roy's Largest Root	20.818	93.682[a]	2.000	9.000	.000
Menu * Depth	Pillai's Trace	1.000	5.117E2	10.000	1.000	.034
	Wilks' Lambda	.000	5.117E2	10.000	1.000	.034
	Hotelling's Trace	5117.433	5.117E2	10.000	1.000	.034
	Roy's Largest Root	5117.433	5.117E2	10.000	1.000	.034

a. Exact statistic

b. Design: Intercept
Within Subjects Design: Menu + Depth + Menu * Depth

Mauchly's Test of Sphericity[b]

Measure:MEASURE_1

Within Subjects Effect	Mauchly's W	Approx. Chi-Square	df	Sig.	Epsilon[a]		
					Greenhouse-Geisser	Huynh-Feldt	Lower-bound
Menu	.029	28.755	14	.014	.441	.571	.200
Depth	.257	12.221	2	.002	.574	.600	.500
Menu * Depth	.000	149.960	54	.000	.293	.428	.100

Tests the null hypothesis that the error covariance matrix of the orthonormalized transformed dependent variables is proportional to an identity matrix.

a. May be used to adjust the degrees of freedom for the averaged tests of significance. Corrected tests are displayed in the Tests of Within-Subjects Effects table.

b. Design: Intercept
Within Subjects Design: Menu + Depth + Menu * Depth

Tests of Within-Subjects Effects

Measure:MEASURE_1

Source		Type III Sum of Squares	df	Mean Square	F	Sig.
Menu	Sphericity Assumed	3.507E7	5	7014468.704	28.488	.000
	Greenhouse-Geisser	3.507E7	2.206	1.590E7	28.488	.000
	Huynh-Feldt	3.507E7	2.856	1.228E7	28.488	.000
	Lower-bound	3.507E7	1.000	3.507E7	28.488	.000
Error(Menu)	Sphericity Assumed	1.231E7	50	246228.108		
	Greenhouse-Geisser	1.231E7	22.056	558178.652		
	Huynh-Feldt	1.231E7	28.562	431045.818		
	Lower-bound	1.231E7	10.000	1231140.539		
Depth	Sphericity Assumed	7.393E7	2	3.697E7	172.411	.000
	Greenhouse-Geisser	7.393E7	1.148	6.443E7	172.411	.000
	Huynh-Feldt	7.393E7	1.200	6.161E7	172.411	.000
	Lower-bound	7.393E7	1.000	7.393E7	172.411	.000
Error(Depth)	Sphericity Assumed	4288227.564	20	214411.378		
	Greenhouse-Geisser	4288227.564	11.476	373674.316		

Tests of Within-Subjects Effects

Measure:MEASURE_1

Source		Type III Sum of Squares	df	Mean Square	F	Sig.
Error(Depth)	Huynh-Feldt	4288227.564	12.001	357334.544		
	Lower-bound	4288227.564	10.000	428822.756		
Menu * Depth	Sphericity Assumed	7993323.997	10	799332.400	8.898	.000
	Greenhouse-Geisser	7993323.997	2.932	2726645.684	8.898	.000
	Huynh-Feldt	7993323.997	4.279	1867955.405	8.898	.000
	Lower-bound	7993323.997	1.000	7993323.997	8.898	.014
Error(Menu*Depth)	Sphericity Assumed	8983589.681	100	89835.897		
	Greenhouse-Geisser	8983589.681	29.316	306444.053		
	Huynh-Feldt	8983589.681	42.792	209937.004		
	Lower-bound	8983589.681	10.000	898358.968		

Tests of Within-Subjects Contrasts

Measure:MEASURE_1

Source	Menu	Depth	Type III Sum of Squares	df	Mean Square	F	Sig.
Menu	Linear	Depth	2198199.277	1	2198199.277	19.583	.001
	Quadratic	Depth	1.799E7	1	1.799E7	55.357	.000
	Cubic	Depth	8798780.683	1	8798780.683	40.481	.000
	Order 4	Depth	1148550.789	1	1148550.789	4.756	.054
	Order 5	Depth	4939380.175	1	4939380.175	14.740	.003
Error(Menu)	Linear	Depth	1122491.552	10	112249.155		
	Quadratic	Depth	3249341.839	10	324934.184		
	Cubic	Depth	2173533.492	10	217353.349		
	Order 4	Depth	2414965.199	10	241496.520		
	Order 5	Depth	3351073.306	10	335107.331		
Depth	Menu * Depth	Linear	7.378E7	1	7.378E7	185.306	.000
		Quadratic	157096.028	1	157096.028	5.119	.047
Error(Depth)	Menu * Depth	Linear	3981350.779	10	398135.078		
		Quadratic	306876.785	10	30687.679		
Menu * Depth	Linear	Linear	206815.826	1	206815.826	5.051	.048
		Quadratic	49.235	1	49.235	.003	.955
	Quadratic	Linear	3517027.141	1	3517027.141	22.156	.001
		Quadratic	374990.364	1	374990.364	7.715	.020
	Cubic	Linear	2130171.599	1	2130171.599	44.043	.000
		Quadratic	124226.393	1	124226.393	2.361	.155
	Order 4	Linear	421607.816	1	421607.816	7.869	.019
		Quadratic	9580.031	1	9580.031	.114	.743
	Order 5	Linear	1184694.487	1	1184694.487	5.785	.037

Tests of Within-Subjects Contrasts

Measure:MEASURE_1

Source	Menu	Depth	Type III Sum of Squares	df	Mean Square	F	Sig.
Menu * Depth	Order 5	Quadratic	24161.103	1	24161.103	.126	.730
Error(Menu*Depth)	Linear	Linear	409484.918	10	40948.492		
		Quadratic	147297.864	10	14729.786		
	Quadratic	Linear	1587391.475	10	158739.147		
		Quadratic	486042.422	10	48604.242		
	Cubic	Linear	483654.585	10	48365.459		
		Quadratic	526059.110	10	52605.911		
	Order 4	Linear	535774.728	10	53577.473		
		Quadratic	840439.142	10	84043.914		
	Order 5	Linear	2047851.277	10	204785.128		
		Quadratic	1919594.161	10	191959.416		

Tests of Between-Subjects Effects

Measure:MEASURE_1
Transformed Variable:Average

Source	Type III Sum of Squares	df	Mean Square	F	Sig.
Intercept	1.059E9	1	1.059E9	370.607	.000
Error	2.857E7	10	2856558.569		

Estimated Marginal Means

1. Menu

Estimates

Measure:MEASURE_1

Menu	Mean	Std. Error	95% Confidence Interval Lower Bound	95% Confidence Interval Upper Bound
1	1930.969	99.437	1709.409	2152.528
2	2527.192	162.339	2165.478	2888.906
3	3133.548	234.820	2610.336	3656.761
4	2276.527	124.286	1999.602	2553.453
5	2046.994	107.901	1806.576	2287.411
6	1958.619	73.471	1794.915	2122.324

Pairwise Comparisons

Measure: MEASURE_1

(I) Menu	(J) Menu	Mean Difference (I-J)	Std. Error	Sig.[a]	95% Confidence Interval for Difference[a]	
					Lower Bound	Upper Bound
1	2	-596.223*	105.656	.003	-1000.605	-191.841
	3	-1202.580*	169.235	.000	-1850.300	-554.859
	4	-345.559*	63.587	.004	-588.926	-102.191
	5	-116.025	60.806	1.000	-348.750	116.700
	6	-27.651	38.883	1.000	-176.467	121.166
2	1	596.223*	105.656	.003	191.841	1000.605
	3	-606.357	163.297	.060	-1231.349	18.636
	4	250.665	93.279	.342	-106.345	607.674
	5	480.198*	112.482	.025	49.693	910.703
	6	568.573*	118.100	.011	116.565	1020.581
3	1	1202.580*	169.235	.000	554.859	1850.300
	2	606.357	163.297	.060	-18.636	1231.349
	4	857.021*	161.621	.005	238.441	1475.601
	5	1086.555*	184.806	.002	379.239	1793.870
	6	1174.929*	182.785	.001	475.349	1874.510
4	1	345.559*	63.587	.004	102.191	588.926
	2	-250.665	93.279	.342	-607.674	106.345
	3	-857.021*	161.621	.005	-1475.601	-238.441
	5	229.533	91.139	.457	-119.288	578.354
	6	317.908*	76.040	.028	26.878	608.938
5	1	116.025	60.806	1.000	-116.700	348.750
	2	-480.198*	112.482	.025	-910.703	-49.693
	3	-1086.555*	184.806	.002	-1793.870	-379.239
	4	-229.533	91.139	.457	-578.354	119.288
	6	88.375	71.044	1.000	-183.536	360.285
6	1	27.651	38.883	1.000	-121.166	176.467
	2	-568.573*	118.100	.011	-1020.581	-116.565
	3	-1174.929*	182.785	.001	-1874.510	-475.349
	4	-317.908*	76.040	.028	-608.938	-26.878
	5	-88.375	71.044	1.000	-360.285	183.536

Based on estimated marginal means

*. The mean difference is significant at the .05 level.

a. Adjustment for multiple comparisons: Bonferroni.

Multivariate Tests

	Value	F	Hypothesis df	Error df	Sig.
Pillai's trace	.886	9.349[a]	5.000	6.000	.008
Wilks' lambda	.114	9.349[a]	5.000	6.000	.008
Hotelling's trace	7.791	9.349[a]	5.000	6.000	.008
Roy's largest root	7.791	9.349[a]	5.000	6.000	.008

Each F tests the multivariate effect of Menu. These tests are based on the linearly independent pairwise comparisons among the estimated marginal means.

a. Exact statistic

2. Depth

Estimates

Measure:MEASURE_1

Depth	Mean	Std. Error	95% Confidence Interval	
			Lower Bound	Upper Bound
1	1584.619	77.905	1411.035	1758.203
2	2272.473	129.196	1984.606	2560.340
3	3079.832	164.370	2713.592	3446.072

Pairwise Comparisons

Measure:MEASURE_1

(I) Depth	(J) Depth	Mean Difference (I-J)	Std. Error	Sig.[a]	95% Confidence Interval for Difference[a]	
					Lower Bound	Upper Bound
1	2	-687.854*	65.070	.000	-874.610	-501.098
	3	-1495.213*	109.839	.000	-1810.460	-1179.966
2	1	687.854*	65.070	.000	501.098	874.610
	3	-807.359*	56.508	.000	-969.540	-645.178
3	1	1495.213*	109.839	.000	1179.966	1810.460
	2	807.359*	56.508	.000	645.178	969.540

Based on estimated marginal means

*. The mean difference is significant at the .05 level.

a. Adjustment for multiple comparisons: Bonferroni.

Multivariate Tests

	Value	F	Hypothesis df	Error df	Sig.
Pillai's trace	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

Each F tests the multivariate effect of Depth. These tests are based on the linearly independent pairwise comparisons among the estimated marginal means.

a. Exact statistic

Multivariate Tests

	Value	F	Hypothesis df	Error df	Sig.
Wilks' lambda	.046	93.682[a]	2.000	9.000	.000
Hotelling's trace	20.818	93.682[a]	2.000	9.000	.000
Roy's largest root	20.818	93.682[a]	2.000	9.000	.000

Each F tests the multivariate effect of Depth. These tests are based on the linearly independent pairwise comparisons among the estimated marginal means.

a. Exact statistic

3. Menu * Depth

Measure:MEASURE_1

Menu	Depth	Mean	Std. Error	95% Confidence Interval	
				Lower Bound	Upper Bound
1	1	1459.603	83.978	1272.489	1646.717
	2	1849.224	101.706	1622.610	2075.839
	3	2484.079	126.107	2203.096	2765.062
2	1	1737.294	90.218	1536.276	1938.312
	2	2433.439	162.807	2070.683	2796.196
	3	3410.842	249.515	2854.887	3966.798
3	1	1891.821	137.591	1585.250	2198.392
	2	3167.058	304.971	2487.539	3846.576
	3	4341.767	354.741	3551.354	5132.180
4	1	1534.327	79.399	1357.415	1711.240
	2	2313.152	158.758	1959.417	2666.886
	3	2982.103	169.090	2605.347	3358.859
5	1	1448.445	78.273	1274.043	1622.848
	2	2066.652	115.041	1810.324	2322.979
	3	2625.885	153.287	2284.339	2967.430
6	1	1436.224	63.399	1294.963	1577.486
	2	1805.315	91.646	1601.114	2009.516
	3	2634.318	94.356	2424.080	2844.557

A.2 EXPERIMENT2: SEARCH TASK

General Linear Model

Within-Subjects Factors

Measure:MEASURE_1

Menu	Device	Dependent Variable
1	1	AAMUS.mouse
	2	AAMUS.stylus
	3	AAMUS.TouchPad
2	1	Default.mouse
	2	Default.stylus
	3	Default.TouchPad
3	1	EMUS.mouse
	2	EMUS.stylus
	3	EMUS.TouchPad

Multivariate Tests[b]

Effect		Value	F	Hypothesis df	Error df	Sig.
Menu	Pillai's Trace	.725	22.391[a]	2.000	17.000	.000
	Wilks' Lambda	.275	22.391[a]	2.000	17.000	.000
	Hotelling's Trace	2.634	22.391[a]	2.000	17.000	.000
	Roy's Largest Root	2.634	22.391[a]	2.000	17.000	.000
Device	Pillai's Trace	.922	1.007E2	2.000	17.000	.000
	Wilks' Lambda	.078	1.007E2	2.000	17.000	.000
	Hotelling's Trace	11.844	1.007E2	2.000	17.000	.000
	Roy's Largest Root	11.844	1.007E2	2.000	17.000	.000
Menu * Device	Pillai's Trace	.359	2.097[a]	4.000	15.000	.132
	Wilks' Lambda	.641	2.097[a]	4.000	15.000	.132
	Hotelling's Trace	.559	2.097[a]	4.000	15.000	.132
	Roy's Largest Root	.559	2.097[a]	4.000	15.000	.132

a. Exact statistic

b. Design: Intercept
Within Subjects Design: Menu + Device + Menu * Device

Mauchly's Test of Sphericity[b]

Measure:MEASURE_1

Within Subjects Effect	Mauchly's W	Approx. Chi-Square	df	Sig.	Epsilon[a]		
					Greenhouse-Geisser	Huynh-Feldt	Lower-bound
Menu	.703	5.980	2	.050	.771	.830	.500
Device	.495	11.963	2	.003	.664	.697	.500
Menu * Device	.686	6.678	9	.672	.835	1.000	.250

Tests the null hypothesis that the error covariance matrix of the orthonormalized transformed dependent variables is proportional to an identity matrix.

a. May be used to adjust the degrees of freedom for the averaged tests of significance. Corrected tests are displayed in the Tests of Within-Subjects Effects table.

b. Design: Intercept
Within Subjects Design: Menu + Device + Menu * Device

Tests of Within-Subjects Effects

Measure:MEASURE_1

Source		Type III Sum of Squares	df	Mean Square	F	Sig.
Menu	Sphericity Assumed	6588413.339	2	3294206.669	11.668	.000
	Greenhouse-Geisser	6588413.339	1.543	4271171.599	11.668	.001
	Huynh-Feldt	6588413.339	1.659	3970568.298	11.668	.000
	Lower-bound	6588413.339	1.000	6588413.339	11.668	.003
Error(Menu)	Sphericity Assumed	1.016E7	36	282329.799		
	Greenhouse-Geisser	1.016E7	27.766	366060.524		

Page 1

Raw statistics from SPSS for Experiment 2 (Search task with 3 input devices).
Summary of Repeated measures Analysis for completion times.

Tests of Within-Subjects Effects

Measure:MEASURE_1

Source		Type III Sum of Squares	df	Mean Square	F	Sig.
Error(Menu)	Huynh-Feldt	1.016E7	29.868	340297.335		
	Lower-bound	1.016E7	18.000	564659.598		
Device	Sphericity Assumed	8.496E7	2	4.248E7	63.026	.000
	Greenhouse-Geisser	8.496E7	1.329	6.394E7	63.026	.000
	Huynh-Feldt	8.496E7	1.394	6.093E7	63.026	.000
	Lower-bound	8.496E7	1.000	8.496E7	63.026	.000
Error(Device)	Sphericity Assumed	2.426E7	36	673996.434		
	Greenhouse-Geisser	2.426E7	23.916	1014535.650		
	Huynh-Feldt	2.426E7	25.097	966784.460		
	Lower-bound	2.426E7	18.000	1347992.867		
Menu * Device	Sphericity Assumed	1064333.805	4	266083.451	1.510	.208
	Greenhouse-Geisser	1064333.805	3.340	318630.982	1.510	.218
	Huynh-Feldt	1064333.805	4.000	266083.451	1.510	.208
	Lower-bound	1064333.805	1.000	1064333.805	1.510	.235
Error(Menu*Device)	Sphericity Assumed	1.269E7	72	176233.447		
	Greenhouse-Geisser	1.269E7	60.126	211036.936		
	Huynh-Feldt	1.269E7	72.000	176233.447		
	Lower-bound	1.269E7	18.000	704933.787		

Tests of Within-Subjects Contrasts

Measure:MEASURE_1

Source	Menu	Device	Type III Sum of Squares	df	Mean Square	F	Sig.
Menu	Linear	Device	4362440.972	1	4362440.972	13.120	.002
	Quadratic	Device	2225972.366	1	2225972.366	9.589	.006
Error(Menu)	Linear	Device	5985203.662	18	332511.315		
	Quadratic	Device	4178669.112	18	232148.284		
Device	Menu * Device	Linear	8.307E7	1	8.307E7	129.465	.000
		Quadratic	1888137.036	1	1888137.036	2.673	.119
Error(Device)	Menu * Device	Linear	1.155E7	18	641641.239		
		Quadratic	1.271E7	18	706351.629		
Menu * Device	Linear	Linear	24007.144	1	24007.144	.118	.735
		Quadratic	512532.928	1	512532.928	3.950	.062
	Quadratic	Linear	394590.572	1	394590.572	1.801	.196
		Quadratic	133203.161	1	133203.161	.871	.363
Error(Menu*Device)	Linear	Linear	3656628.629	18	203146.035		
		Quadratic	2335705.528	18	129761.418		
	Quadratic	Linear	3943708.464	18	219094.804		
		Quadratic	2752767.537	18	152931.530		

Tests of Between-Subjects Effects

Measure:MEASURE_1
Transformed Variable:Average

Source	Type III Sum of Squares	df	Mean Square	F	Sig.
Intercept	4.618E9	1	4.618E9	4936.535	.000
Error	1.684E7	18	935507.038		

Estimated Marginal Means

1. Menu

Estimates

Measure:MEASURE_1

Menu	Mean	Std. Error	95% Confidence Interval	
			Lower Bound	Upper Bound
1	5311.750	83.451	5136.427	5487.074

Estimates

Measure:MEASURE_1

Menu	Mean	Std. Error	95% Confidence Interval	
			Lower Bound	Upper Bound
2	5358.160	94.727	5159.146	5557.174
3	4920.511	101.889	4706.449	5134.573

Pairwise Comparisons

Measure:MEASURE_1

(I) Menu	(J) Menu	Mean Difference (I-J)	Std. Error	Sig.[a]	95% Confidence Interval for Difference[a]	
					Lower Bound	Upper Bound
1	2	-46.410	115.940	1.000	-352.393	259.573
	3	391.239*	108.014	.006	106.174	676.304
2	1	46.410	115.940	1.000	-259.573	352.393
	3	437.649*	67.895	.000	258.464	616.834
3	1	-391.239*	108.014	.006	-676.304	-106.174
	2	-437.649*	67.895	.000	-616.834	-258.464

Based on estimated marginal means

a. Adjustment for multiple comparisons: Bonferroni.

*. The mean difference is significant at the .05 level.

Multivariate Tests

	Value	F	Hypothesis df	Error df	Sig.
Pillai's trace	.725	22.391[a]	2.000	17.000	.000
Wilks' lambda	.275	22.391[a]	2.000	17.000	.000
Hotelling's trace	2.634	22.391[a]	2.000	17.000	.000
Roy's largest root	2.634	22.391[a]	2.000	17.000	.000

Each F tests the multivariate effect of Menu. These tests are based on the linearly independent pairwise comparisons among the estimated marginal means.

a. Exact statistic

2. Device

Estimates

Measure:MEASURE_1

Device	Mean	Std. Error	95% Confidence Interval	
			Lower Bound	Upper Bound
1	4417.478	59.609	4292.245	4542.711
2	5048.202	95.655	4847.239	5249.165
3	6124.741	165.404	5777.241	6472.242

Pairwise Comparisons

Measure:MEASURE_1

(I) Device	(J) Device	Mean Difference (I-J)	Std. Error	Sig.[a]	95% Confidence Interval for Difference[a]	
					Lower Bound	Upper Bound
1	2	-630.724*	98.448	.000	-890.542	-370.906
	3	-1707.264*	150.046	.000	-2103.256	-1311.271
2	1	630.724*	98.448	.000	370.906	890.542
	3	-1076.539*	196.828	.000	-1595.997	-557.082
3	1	1707.264*	150.046	.000	1311.271	2103.256
	2	1076.539*	196.828	.000	557.082	1595.997

Based on estimated marginal means

*. The mean difference is significant at the .05 level.

a. Adjustment for multiple comparisons: Bonferroni.

Multivariate Tests

	Value	F	Hypothesis df	Error df	Sig.
Pillai's trace	.922	1.007E2	2.000	17.000	.000
Wilks' lambda	.078	1.007E2	2.000	17.000	.000
Hotelling's trace	11.844	1.007E2	2.000	17.000	.000
Roy's largest root	11.844	1.007E2	2.000	17.000	.000

Each F tests the multivariate effect of Device. These tests are based on the linearly independent pairwise comparisons among the estimated marginal means.

a. Exact statistic

3. Menu * Device

Measure:MEASURE_1

Menu	Device	Mean	Std. Error	95% Confidence Interval	
				Lower Bound	Upper Bound
1	1	4522.792	80.951	4352.721	4692.862
	2	5230.060	138.993	4938.046	5522.074
	3	6182.399	180.071	5804.083	6560.715
2	1	4467.718	102.085	4253.246	4682.190
	2	5265.375	117.636	5018.230	5512.520
	3	6341.387	178.255	5966.886	6715.888
3	1	4261.924	86.648	4079.883	4443.965
	2	4649.171	115.149	4407.251	4891.091
	3	5850.439	227.827	5371.792	6329.086

A.3 EXPERIMENT3: OBSERVATION STUDY

T-Test

Paired Samples Statistics

		Mean	N	Std. Deviation	Std. Error Mean
Pair 1	AAMUS	1649.7333	9	305.09374	101.69791
	Default	1716.2815	9	269.84678	89.94893

Paired Samples Correlations

		N	Correlation	Sig.
Pair 1	AAMUS & Default	9	.420	.260

Paired Samples Test

		Paired Differences					
					95% Confidence Interval of the Difference		
		Mean	Std. Deviation	Std. Error Mean	Lower	Upper	t
Pair 1	AAMUS - Default	-66.54815	310.97831	103.65944	-305.58724	172.49094	-.642

Paired Samples Test

		Paired Differences	
		df	Sig. (2-tailed)
Pair 1	AAMUS - Default	8	.539

Page 1

Raw statistics from SPSS for trapped trials from observation study. Summary of t-test for completion times in experiment 3.

T-Test

Paired Samples Statistics

		Mean	N	Std. Deviation	Std. Error Mean
Pair 1	AAMUS	1280.6074	9	193.92858	64.64286
	Default	1539.5630	9	273.11379	91.03793

Paired Samples Correlations

		N	Correlation	Sig.
Pair 1	AAMUS & Default	9	.821	.007

Paired Samples Test

		Paired Differences					
					95% Confidence Interval of the Difference		
		Mean	Std. Deviation	Std. Error Mean	Lower	Upper	t
Pair 1	AAMUS - Default	-258.95556	158.88454	52.96151	-381.08502	-136.82609	-4.890

Paired Samples Test

		Paired Differences	
		df	Sig. (2-tailed)
Pair 1	AAMUS - Default	8	.001

Page 1

Raw statistics from SPSS for clear trials from observation study. Summary of t-test for completion times in experiment 3.

A.4 EXPERIMENT4: TESTING ALTERNATE DESIGNS

Univariate Analysis of Variance: for select task

Tests of Between-Subjects Effects

Dependent Variable:Completion time

Source		Type III Sum of Squares	df	Mean Square	F	Sig.
Intercept	Hypothesis	1.216E9	1	1.216E9	864.362	.000
	Error	1.547E7	11.000	1.406E6		
Menu_name	Hypothesis	970039.194	3	323346.398	2.524	.075
	Error	4232252.198	33.034	128116.813[b]		

Tests of Between-Subjects Effects

Dependent Variable:Completion time

Source		Type III Sum of Squares	df	Mean Square	F	Sig.
ParticipentNo	Hypothesis	1.548E7	11	1407109.848	10.980	.000
	Error	4231847.291	33.021	128154.731[c]		
Menu_name * ParticipentNo	Hypothesis	4231178.744	33	128217.538	2.513	.000
	Error	4.536E7	889	51022.912[d]		

a. .999 MS(ParticipentNo) + .001 MS(Error)

b. .999 MS(Menu_name * ParticipentNo) + .001 MS(Error)

c. .999 MS(Menu_name * ParticipentNo) + .001 MS(Error)

d. MS(Error)

Expected Mean Squares[a,b]

Source	Variance Component			
	Var (Participent No)	Var (Menu_name * ParticipentNo)	Var(Error)	Quadratic Term
Intercept	77.936	19.484	1.000	Intercept, Menu_name
Menu_name	.000	19.485	1.000	Menu_name
ParticipentNo	77.976	19.494	1.000	
Menu_name * ParticipentNo	.000	19.510	1.000	
Error	.000	.000	1.000	

a. For each source, the expected mean square equals the sum of the coefficients in the cells times the variance components, plus a quadratic term involving effects in the Quadratic Term cell.

b. Expected Mean Squares are based on the Type III Sums of Squares.

Estimated Marginal Means

Menu_name

Estimates

Dependent Variable:Completion time

Menu_name	Mean	Std. Error	95% Confidence Interval	
			Lower Bound	Upper Bound
AAMUS	1108.583	14.817	1079.502	1137.664
Click	1125.846	14.810	1096.780	1154.912
Curvy	1132.147	14.712	1103.274	1161.020
Hover	1193.833	14.751	1164.882	1222.783

Pairwise Comparisons

Dependent Variable:Completion time

(I) Menu_name	(J) Menu_name	Mean Difference (I-J)	Std. Error	Sig.[a]	95% Confidence Interval for Difference[a]	
					Lower Bound	Upper Bound
AAMUS	Click	-17.263	20.950	1.000	-72.657	38.132
	Curvy	-23.564	20.880	1.000	-78.775	31.647
	Hover	-85.249*	20.908	.000	-140.534	-29.965

Based on estimated marginal means

a. Adjustment for multiple comparisons: Bonferroni.

*. The mean difference is significant at the .05 level.

Raw statistics from SPSS from experiment 4, testing the alternate designs.

Pairwise Comparisons

Dependent Variable:Completion_time

(I) Menu_name	(J) Menu_name	Mean Difference (I-J)	Std. Error	Sig.[a]	95% Confidence Interval for Difference[a] Lower Bound	Upper Bound
Click	AAMUS	17.263	20.950	1.000	-38.132	72.657
	Curvy	-6.301	20.875	1.000	-61.498	48.896
	Hover	-67.987*	20.903	.007	-123.257	-12.717
Curvy	AAMUS	23.564	20.880	1.000	-31.647	78.775
	Click	6.301	20.875	1.000	-48.896	61.498
	Hover	-61.686*	20.833	.019	-116.772	-6.600
Hover	AAMUS	85.249*	20.908	.000	29.965	140.534
	Click	67.987*	20.903	.007	12.717	123.257
	Curvy	61.686*	20.833	.019	6.600	116.772

Based on estimated marginal means

a. Adjustment for multiple comparisons: Bonferroni.

*. The mean difference is significant at the .05 level.

Univariate Tests

Dependent Variable:Completion_time

	Sum of Squares	df	Mean Square	F	Sig.
Contrast	970039.194	3	323346.398	6.337	.000
Error	4.536E7	889	51022.912		

The F tests the effect of Menu_name. This test is based on the linearly independent pairwise comparisons among the estimated marginal means.

Post Hoc Tests

Menu_name

Multiple Comparisons

Completion_time
Tamhane

(I) Menu_name	(J) Menu_name	Mean Difference (I-J)	Std. Error	Sig.	95% Confidence Interval Lower Bound	Upper Bound
AAMUS	Click	-14.83	22.712	.987	-74.84	45.19
	Curvy	-25.07	23.320	.864	-86.69	36.55
	Hover	-84.89*	25.127	.005	-151.29	-18.48
Click	AAMUS	14.83	22.712	.987	-45.19	74.84
	Curvy	-10.24	23.579	.999	-72.54	52.06
	Hover	-70.06*	25.368	.035	-137.09	-3.02
Curvy	AAMUS	25.07	23.320	.864	-36.55	86.69
	Click	10.24	23.579	.999	-52.06	72.54
	Hover	-59.81	25.913	.122	-128.29	8.66
Hover	AAMUS	84.89*	25.127	.005	18.48	151.29
	Click	70.06*	25.368	.035	3.02	137.09
	Curvy	59.81	25.913	.122	-8.66	128.29

Based on observed means.
The error term is Mean Square(Error) = 51022.912.

*. The mean difference is significant at the .05 level.

Summary of ANOVA for completion times among menu types for select task.

Univariate Analysis of Variance: for search task

Tests of Between-Subjects Effects

Dependent Variable:Completion_time

Source		Type III Sum of Squares	df	Mean Square	F	Sig.
Intercept	Hypothesis	3.346E9	1	3.346E9	180.065	.000
	Error	2.235E8	12.030	1.858E7		
Menu_name	Hypothesis	1.486E7	3	4953055.620	3.743	.019
	Error	4.938E7	37.312	1.323E6		
ParticipentNo	Hypothesis	2.354E8	12	1.962E7	14.276	.000
	Error	4.947E7	36.001	1.374E6		
Menu_name * ParticipentNo	Hypothesis	4.947E7	36	1374115.432	3.161	.000
	Error	4.978E8	1145	434766.073[d]		

a. .946 MS(ParticipentNo) + .054 MS(Error)

b. .946 MS(Menu_name * ParticipentNo) + .054 MS(Error)

c. 1.000 MS(Menu_name * ParticipentNo) + 4.91E-005 MS(Error)

d. MS(Error)

Expected Mean Squares[a,b]

Source	Variance Component			
	Var (Participent No)	Var (Menu_name * ParticipentNo)	Var(Error)	Quadratic Term
Intercept	86.383	21.596	1.000	Intercept, Menu_name
Menu_name	.000	21.596	1.000	Menu_name
ParticipentNo	91.310	22.828	1.000	
Menu_name * ParticipentNo	.000	22.829	1.000	
Error	.000	.000	1.000	

a. For each source, the expected mean square equals the sum of the coefficients in the cells times the variance components, plus a quadratic term involving effects in the Quadratic Term cell.

b. Expected Mean Squares are based on the Type III Sums of Squares.

Estimated Marginal Means

Menu_name

Estimates

Dependent Variable:Completion_time

Menu_name	Mean	Std. Error	95% Confidence Interval Lower Bound	95% Confidence Interval Upper Bound
AAMUS	1664.867	39.374	1587.614	1742.121
Click	1756.709	39.374	1679.456	1833.963
Curvy	1586.913	39.288	1509.829	1663.998
Hover	1895.928	39.374	1818.675	1973.182

Pairwise Comparisons

Dependent Variable:Completion_time

(I) Menu_name	(J) Menu_name	Mean Difference (I-J)	Std. Error	Sig.[a]	95% Confidence Interval for Difference[a] Lower Bound	95% Confidence Interval for Difference[a] Upper Bound
AAMUS	Click	-91.842	55.683	.596	-239.005	55.321
	Curvy	77.954	55.623	.968	-69.048	224.956
	Hover	-231.061*	55.683	.000	-378.224	-83.898
Click	AAMUS	91.842	55.683	.596	-55.321	239.005
	Curvy	169.796*	55.623	.014	22.794	316.798
	Hover	-139.219	55.683	.075	-286.382	7.944
Curvy	AAMUS	-77.954	55.623	.968	-224.956	69.048
	Click	-169.796*	55.623	.014	-316.798	-22.794
	Hover	-309.015*	55.623	.000	-456.017	-162.013
Hover	AAMUS	231.061*	55.683	.000	83.898	378.224
	Click	139.219	55.683	.075	-7.944	286.382
	Curvy	309.015*	55.623	.000	162.013	456.017

Based on estimated marginal means

a. Adjustment for multiple comparisons: Bonferroni.

*. The mean difference is significant at the .05 level.

Univariate Tests

Dependent Variable:Completion_time

	Sum of Squares	df	Mean Square	F	Sig.
Contrast	1.486E7	3	4953055.620	11.392	.000
Error	4.978E8	1145	434766.073		

The F tests the effect of Menu_name. This test is based on the linearly independent pairwise comparisons among the estimated marginal means.

Post Hoc Tests

Menu_name

Multiple Comparisons

Completion_time
Tamhane

(I) Menu_name	(J) Menu_name	Mean Difference (I-J)	Std. Error	Sig.	95% Confidence Interval	
					Lower Bound	Upper Bound
AAMUS	Click	-83.10	63.310	.717	-250.25	84.04
	Curvy	57.53	55.913	.886	-90.07	205.13
	Hover	-214.82*	69.581	.013	-398.57	-31.07
Click	AAMUS	83.10	63.310	.717	-84.04	250.25
	Curvy	140.63	62.447	.139	-24.24	305.50
	Hover	-131.72	74.932	.391	-329.54	66.10
Curvy	AAMUS	-57.53	55.913	.886	-205.13	90.07
	Click	-140.63	62.447	.139	-305.50	24.24
	Hover	-272.35*	68.796	.001	-454.03	-90.66
Hover	AAMUS	214.82*	69.581	.013	31.07	398.57

Based on observed means.
The error term is Mean Square(Error) = 434766.073.

Multiple Comparisons

Completion_time
Tamhane

(I) Menu_name	(J) Menu_name	Mean Difference (I-J)	Std. Error	Sig.	95% Confidence Interval	
					Lower Bound	Upper Bound
Hover	Click	131.72	74.932	.391	-66.10	329.54
	Curvy	272.35*	68.796	.001	90.66	454.03

Based on observed means.
The error term is Mean Square(Error) = 434766.073.

*. The mean difference is significant at the .05 level.

Summary of ANOVA for completion times among menu types for search task.

A.5 EXPERIMENT5: FINAL STUDY

Means

[DataSet1] E:\HCI_LAB_CODE\Erum_SVN\experiment_Results\March2009\March2\b\final_all.sav

Case Processing Summary

	Cases					
	Included		Excluded		Total	
	N	Percent	N	Percent	N	Percent
Completion_time * Menu_name	16801	100.0%	0	.0%	16801	100.0%
Completion_time * Device	16801	100.0%	0	.0%	16801	100.0%
Completion_time * Search/Select	16801	100.0%	0	.0%	16801	100.0%
Completion_time * Ratio	16801	100.0%	0	.0%	16801	100.0%

Completion_time * Menu_name

Completion_time

Menu_name	Mean	N	Std. Deviation
AAMUS	1746.98	4197	730.013
AAMUS_Click	1727.94	4214	727.385
AAMUS_Curvy	1749.98	4218	727.367
Default	2024.97	4172	794.146
Total	1811.99	16801	755.131

Completion_time * Device

Completion_time

Device	Mean	N	Std. Deviation
mouse	1459.27	10405	521.202
touch_pad	2385.78	6396	724.117
Total	1811.99	16801	755.131

Completion_time * Search/Select

Completion_time

Se...	Mean	N	Std. Deviation
0	1611.28	8420	671.229
1	2013.63	8381	780.629
Total	1811.99	16801	755.131

Completion_time * Ratio

Completion_time

Ratio	Mean	N	Std. Deviation
60	1841.68	10223	762.979
75	1765.85	6578	740.468
Total	1811.99	16801	755.131

Page 1

Summary of mean completion times from experiment 5 for all menu types, device types and task types.

Univariate Analysis of Variance

Between-Subjects Factors

		N
Menu_name	AAMUS	1302
	AAMUS_Click	1306
	AAMUS_Curvy	1304
	Default	1293
Ratio	60	3169
	75	2036
Participent No	1	160
	2	163
	3	162
	4	162
	5	163
	6	164
	7	164
	8	161
	9	164
	10	163
	11	160
	12	163
	13	164
	14	161
	15	164
	16	160
	38	163
	39	164
	41	158
	42	163
	43	164
	44	164
	45	163
	46	163
	47	164
	48	164
	49	162
	50	163
	51	164
	52	161
	53	163
	54	164

Tests of Between-Subjects Effects

Dependent Variable:Completion_time

Source		Type III Sum of Squares	df	Mean Square	F	Sig.
Intercept	Hypothesis	8.060E9	1	8.060E9	2739.803	.000
	Error	9.119E7	31.000	2.942E6		
Menu_name	Hypothesis	7.249E7	3	2.416E7	82.592	.000
	Error	2.721E7	93.020	292549.378 [b]		
Ratio	Hypothesis	5073843.908	1	5073843.908	31.236	.000
	Error	5036366.186	31.005	162434.731 [c]		
ParticipentNo	Hypothesis	9.120E7	31	2941982.667	13.071	.000
	Error	4873996.282	21.654	225084.397 [d]		
Menu_name * Ratio	Hypothesis	879387.507	3	293129.169	1.275	.288
	Error	2.139E7	93.026	229905.551 [e]		
Menu_name * ParticipentNo	Hypothesis	2.721E7	93	292577.005	1.273	.124
	Error	2.138E7	93	229920.412 [f]		
Ratio * ParticipentNo	Hypothesis	5035492.109	31	162435.229	.707	.863
	Error	2.139E7	93.015	229911.767 [g]		
Menu_name * Ratio * ParticipentNo	Hypothesis	2.138E7	93	229920.412	1.465	.003
	Error	7.769E8	4949	156983.220 [h]		

a. 1.000 MS(ParticipentNo) + 9.14E-005 MS(Error)

b. 1.000 MS(Menu_name * ParticipentNo) + .000 MS(Error)

c. 1.000 MS(Ratio * ParticipentNo) + 9.14E-005 MS(Error)

d. 1.000 MS(Menu_name * ParticipentNo) + MS(Ratio * ParticipentNo) - 1.000 MS(Menu_name * Ratio * ParticipentNo)

e. 1.000 MS(Menu_name * Ratio * ParticipentNo) + .000 MS(Error)

f. MS(Menu_name * Ratio * ParticipentNo)

g. 1.000 MS(Menu_name * Ratio * ParticipentNo) + .000 MS(Error)

h. MS(Error)

Expected Mean Squares[a,b]

Source	Variance Component					
	Var (ParticipentNo)	Var (Menu_name * ParticipentNo)	Var(Ratio * ParticipentNo)	Var (Menu_name * Ratio * ParticipentNo)	Var(Error)	Quadratic Term
Intercept	154.867	38.717	77.434	19.358	1.000	Intercept, Menu_name, Ratio, Menu_name * Ratio
Menu_name	.000	38.717	.000	19.359	1.000	Menu_name, Menu_name * Ratio
Ratio	.000	.000	77.434	19.358	1.000	Ratio, Menu_name * Ratio
ParticipentNo	154.881	38.720	77.441	19.360	1.000	
Menu_name * Ratio	.000	.000	.000	19.359	1.000	Menu_name * Ratio
Menu_name * ParticipentNo	.000	38.725	.000	19.362	1.000	
Ratio * ParticipentNo	.000	.000	77.441	19.360	1.000	
Menu_name * Ratio * ParticipentNo	.000	.000	.000	19.362	1.000	
Error	.000	.000	.000	.000	1.000	

a. For each source, the expected mean square equals the sum of the coefficients in the cells times the variance components, plus a quadratic term involving effects in the Quadratic Term cell.

b. Expected Mean Squares are based on the Type III Sums of Squares.

Estimated Marginal Means

1. Grand Mean

Dependent Variable:Completion_time

Mean	Std. Error	95% Confidence Interval Lower Bound	95% Confidence Interval Upper Bound
1275.283	5.628	1264.249	1286.316

2. Menu_name

Estimates

Dependent Variable:Completion_time

Menu_name	Mean	Std. Error	95% Confidence Interval Lower Bound	95% Confidence Interval Upper Bound
AAMUS	1201.521	11.255	1179.457	1223.586
AAMUS_Click	1192.340	11.243	1170.299	1214.381
AAMUS_Curvy	1223.051	11.243	1201.010	1245.093
Default	1484.218	11.285	1462.094	1506.341

Pairwise Comparisons

Dependent Variable:Completion_time

(I) Menu_name	(J) Menu_name	Mean Difference (I-J)	Std. Error	Sig.[a]	95% Confidence Interval for Difference[a] Lower Bound	95% Confidence Interval for Difference[a] Upper Bound
AAMUS	AAMUS_Click	9.182	15.908	1.000	-32.805	51.169
	AAMUS_Curvy	-21.530	15.909	1.000	-63.518	20.457
	Default	-282.697*	15.938	.000	-324.762	-240.631
AAMUS_Click	AAMUS	-9.182	15.908	1.000	-51.169	32.805
	AAMUS_Curvy	-30.712	15.900	.321	-72.677	11.254
	Default	-291.878*	15.930	.000	-333.921	-249.835
AAMUS_Curvy	AAMUS	21.530	15.909	1.000	-20.457	63.518
	AAMUS_Click	30.712	15.900	.321	-11.254	72.677
	Default	-261.166*	15.930	.000	-303.210	-219.122
Default	AAMUS	282.697*	15.938	.000	240.631	324.762
	AAMUS_Click	291.878*	15.930	.000	249.835	333.921
	AAMUS_Curvy	261.166*	15.930	.000	219.122	303.210

Based on estimated marginal means

a. Adjustment for multiple comparisons: Bonferroni.

*. The mean difference is significant at the .05 level.

Univariate Tests

Dependent Variable:Completion_time

	Sum of Squares	df	Mean Square	F	Sig.
Contrast	7.249E7	3	2.416E7	153.916	.000
Error	7.769E8	4949	156983.220		

The F tests the effect of Menu_name. This test is based on the linearly independent pairwise comparisons among the estimated marginal means.

Univariate Analysis of Variance

Between-Subjects Factors

		N
Menu_name	AAMUS	1299
	AAMUS_Click	1298
	AAMUS_Curvy	1307
	Default	1296
Ratio	60	3171
	75	2029
Participant No	1	163
	2	163
	3	163
	4	164
	5	164
	6	162
	7	164
	8	162
	9	163
	10	164
	11	160
	12	164
	13	164
	14	163
	15	163
	16	158
	38	161
	39	163
	41	163
	42	164
	43	164
	44	163
	45	158
	46	164
	47	164
	48	162
	49	160
	50	160
	51	164
	52	163
	53	162
	54	161

Tests of Between-Subjects Effects

Dependent Variable:Completion_time

Source		Type III Sum of Squares	df	Mean Square	F	Sig.
Intercept	Hypothesis	1.313E10	1	1.313E10	2666.303	.000
	Error	1.527E8	31.000	4.924E6 [a]		
Menu_name	Hypothesis	6.781E7	3	2.260E7	58.482	.000
	Error	3.596E7	93.030	386503.360 [b]		
Ratio	Hypothesis	1.082E7	1	1.082E7	9.674	.004
	Error	3.468E7	31.002	1.119E6 [c]		
ParticipentNo	Hypothesis	1.527E8	31	4925022.015	4.222	.000
	Error	3.673E7	31.487	1.166E6 [d]		
Menu_name * Ratio	Hypothesis	543204.284	3	181068.095	.534	.660
	Error	3.152E7	93.034	338766.343 [e]		
Menu_name * ParticipentNo	Hypothesis	3.595E7	93	386545.948	1.141	.263
	Error	3.151E7	93	338796.016 [f]		
Ratio * ParticipentNo	Hypothesis	3.468E7	31	1118666.205	3.302	.000
	Error	3.151E7	93.020	338778.465 [g]		
Menu_name * Ratio * ParticipentNo	Hypothesis	3.151E7	93	338796.016	1.479	.002
	Error	1.133E9	4944	229088.382 [h]		

a. 1.000 MS(ParticipentNo) + .000 MS(Error)

b. 1.000 MS(Menu_name * ParticipentNo) + .000 MS(Error)

c. 1.000 MS(Ratio * ParticipentNo) + .000 MS(Error)

d. 1.000 MS(Menu_name * ParticipentNo) + MS(Ratio * ParticipentNo) - 1.000 MS(Menu_name * Ratio * ParticipentNo)

e. 1.000 MS(Menu_name * Ratio * ParticipentNo) + .000 MS(Error)

f. MS(Menu_name * Ratio * ParticipentNo)

g. 1.000 MS(Menu_name * Ratio * ParticipentNo) + .000 MS(Error)

h. MS(Error)

Expected Mean Squares[a,b]

Source	Variance Component					
	Var (Participent No)	Var (Menu_name * ParticipentNo)	Var(Ratio * ParticipentNo)	Var (Menu_name * Ratio * ParticipentNo)	Var(Error)	Quadratic Term
Intercept	154.551	38.638	77.276	19.319	1.000	Intercept, Menu_name, Ratio, Menu_name * Ratio
Menu_name	.000	38.638	.000	19.319	1.000	Menu_name, Menu_name * Ratio
Ratio	.000	.000	77.276	19.319	1.000	Ratio, Menu_name * Ratio
ParticipentNo	154.570	38.642	77.285	19.321	1.000	
Menu_name * Ratio	.000	.000	.000	19.319	1.000	Menu_name * Ratio
Menu_name * ParticipentNo	.000	38.649	.000	19.324	1.000	
Ratio * ParticipentNo	.000	.000	77.285	19.321	1.000	
Menu_name * Ratio * ParticipentNo	.000	.000	.000	19.324	1.000	
Error	.000	.000	.000	.000	1.000	

a. For each source, the expected mean square equals the sum of the coefficients in the cells times the variance components, plus a quadratic term involving effects in the Quadratic Term cell.

b. Expected Mean Squares are based on the Type III Sums of Squares.

Estimated Marginal Means

1. Grand Mean

Dependent Variable:Completion_time

Mean	Std. Error	95% Confidence Interval	
		Lower Bound	Upper Bound
1629.381	6.806	1616.039	1642.724

2. Menu_name

Estimates

Dependent Variable:Completion_time

Menu_name	Mean	Std. Error	95% Confidence Interval	
			Lower Bound	Upper Bound
AAMUS	1586.139	13.612	1559.455	1612.824
AAMUS_Click	1530.300	13.625	1503.590	1557.010
AAMUS_Curvy	1571.720	13.572	1545.113	1598.327
Default	1829.366	13.640	1802.626	1856.106

Pairwise Comparisons

Dependent Variable:Completion_time

(I) Menu_name	(J) Menu_name	Mean Difference (I-J)	Std. Error	Sig.[a]	95% Confidence Interval for Difference[a]	
					Lower Bound	Upper Bound
AAMUS	AAMUS_Click	55.840*	19.259	.023	5.010	106.670
	AAMUS_Curvy	14.419	19.222	1.000	-36.313	65.151
	Default	-243.227*	19.270	.000	-294.086	-192.368
AAMUS_Click	AAMUS	-55.840*	19.259	.023	-106.670	-5.010
	AAMUS_Curvy	-41.421	19.231	.188	-92.177	9.336
	Default	-299.067*	19.279	.000	-349.949	-248.184
AAMUS_Curvy	AAMUS	-14.419	19.222	1.000	-65.151	36.313
	AAMUS_Click	41.421	19.231	.188	-9.336	92.177
	Default	-257.646*	19.242	.000	-308.431	-206.861
Default	AAMUS	243.227*	19.270	.000	192.368	294.086
	AAMUS_Click	299.067*	19.279	.000	248.184	349.949
	AAMUS_Curvy	257.646*	19.242	.000	206.861	308.431

Based on estimated marginal means

*. The mean difference is significant at the .05 level.

a. Adjustment for multiple comparisons: Bonferroni.

Univariate Tests

Dependent Variable:Completion_time

	Sum of Squares	df	Mean Square	F	Sig.
Contrast	6.781E7	3	2.260E7	98.666	.000
Error	1.133E9	4944	229088.382		

The F tests the effect of Menu_name. This test is based on the linearly independent pairwise comparisons among the estimated marginal means.

Univariate Analysis of Variance, touch, select, ratio

Between-Subjects Factors

		N
Menu_name	AAMUS	805
	AAMUS_Click	803
	AAMUS_Curvy	811
	Default	796
Ratio	60	1959
	75	1256
Participent No	18	161
	19	164
	20	160
	21	161
	22	158
	23	162
	24	162
	25	162
	26	163
	27	163
	28	162
	29	155
	30	161
	31	159
	32	159
	33	163
	35	162
	36	155
	37	162
	40	161

Tests of Between-Subjects Effects

Dependent Variable:Completion time

Source		Type III Sum of Squares	df	Mean Square	F	Sig.
Intercept	Hypothesis	1.403E10	1	1.403E10	2165.001	.000
	Error	1.231E8	19.001	6.480E6		
Menu_name	Hypothesis	4.948E7	3	1.649E7	26.979	.000
	Error	3.487E7	57.042	611360.501 [b]		

a. 1.000 MS(ParticipentNo) + .000 MS(Error)

b. .999 MS(Menu_name * ParticipentNo) + .001 MS(Error)

c. 1.000 MS(Ratio * ParticipentNo) + .000 MS(Error)

d. 1.000 MS(Menu_name * ParticipentNo) + MS(Ratio * ParticipentNo) - 1.000 MS(Menu_name * Ratio * ParticipentNo)

e. .999 MS(Menu_name * Ratio * ParticipentNo) + .001 MS(Error)

f. MS(Menu_name * Ratio * ParticipentNo)

g. 1.000 MS(Menu_name * Ratio * ParticipentNo) + .000 MS(Error)

h. MS(Error)

Tests of Between-Subjects Effects

Dependent Variable:Completion time

Source		Type III Sum of Squares	df	Mean Square	F	Sig.
Ratio	Hypothesis	2975384.396	1	2975384.396	2.170	.157
	Error	2.605E7	19.003	1.371E6		
ParticipentNo	Hypothesis	1.231E8	19	6481512.086	4.169	.001
	Error	3.456E7	22.228	1.555E6		
Menu_name * Ratio	Hypothesis	427793.935	3	142597.978	.333	.801
	Error	2.442E7	57.061	428031.167 [e]		
Menu_name * ParticipentNo	Hypothesis	3.486E7	57	611550.779	1.429	.091
	Error	2.440E7	57	428096.098 [f]		
Ratio * ParticipentNo	Hypothesis	2.605E7	19	1371240.709	3.203	.000
	Error	2.442E7	57.039	428054.760 [g]		
Menu_name * Ratio * ParticipentNo	Hypothesis	2.440E7	57	428096.098	1.285	.075
	Error	1.018E9	3055	333063.232 [h]		

a. 1.000 MS(ParticipentNo) + .000 MS(Error)

b. .999 MS(Menu_name * ParticipentNo) + .001 MS(Error)

c. 1.000 MS(Ratio * ParticipentNo) + .000 MS(Error)

d. 1.000 MS(Menu_name * ParticipentNo) + MS(Ratio * ParticipentNo) - 1.000 MS(Menu_name * Ratio * ParticipentNo)

e. .999 MS(Menu_name * Ratio * ParticipentNo) + .001 MS(Error)

f. MS(Menu_name * Ratio * ParticipentNo)

g. 1.000 MS(Menu_name * Ratio * ParticipentNo) + .000 MS(Error)

h. MS(Error)

Expected Mean Squares[a,b]

Source	Variance Component					
	Var (Participent No)	Var (Menu_name * ParticipentNo)	Var(Ratio * ParticipentNo)	Var (Menu_name * Ratio * ParticipentNo)	Var(Error)	Quadratic Term
Intercept	152.789	38.197	76.394	19.099	1.000	Intercept, Menu_name, Ratio, Menu_name * Ratio
Menu_name	.000	38.199	.000	19.100	1.000	Menu_name, Menu_name * Ratio
Ratio	.000	.000	76.394	19.099	1.000	Ratio, Menu_name * Ratio
ParticipentNo	152.835	38.209	76.417	19.104	1.000	
Menu_name * Ratio	.000	.000	.000	19.100	1.000	Menu_name * Ratio
Menu_name * ParticipentNo	.000	38.225	.000	19.113	1.000	
Ratio * ParticipentNo	.000	.000	76.417	19.104	1.000	
Menu_name * Ratio * ParticipentNo	.000	.000	.000	19.113	1.000	
Error	.000	.000	.000	.000	1.000	

a. For each source, the expected mean square equals the sum of the coefficients in the cells times the variance components, plus a quadratic term involving effects in the Quadratic Term cell.

b. Expected Mean Squares are based on the Type III Sums of Squares.

Estimated Marginal Means

2. Menu_name

Estimates

Dependent Variable:Completion_time

Menu_name	Mean	Std. Error	95% Confidence Interval	
			Lower Bound	Upper Bound
AAMUS	2048.576	20.849	2007.698	2089.455
AAMUS_Click	2108.933	20.841	2068.069	2149.797
AAMUS_Curvy	2052.804	20.796	2012.029	2093.579
Default	2360.160	21.034	2318.917	2401.402

Pairwise Comparisons

Dependent Variable:Completion_time

(I) Menu_name	(J) Menu_name	Mean Difference (I-J)	Std. Error	Sig.[a]	95% Confidence Interval for Difference[a]	
					Lower Bound	Upper Bound
AAMUS	AAMUS_Click	-60.356	29.479	.244	-138.181	17.468
	AAMUS_Curvy	-4.227	29.447	1.000	-81.967	73.512
	Default	-311.583*	29.616	.000	-389.768	-233.398
AAMUS_Click	AAMUS	60.356	29.479	.244	-17.468	138.181
	AAMUS_Curvy	56.129	29.442	.340	-21.597	133.854
	Default	-251.227*	29.611	.000	-329.398	-173.056
AAMUS_Curvy	AAMUS	4.227	29.447	1.000	-73.512	81.967
	AAMUS_Click	-56.129	29.442	.340	-133.854	21.597
	Default	-307.356*	29.579	.000	-385.443	-229.269
Default	AAMUS	311.583*	29.616	.000	233.398	389.768
	AAMUS_Click	251.227*	29.611	.000	173.056	329.398
	AAMUS_Curvy	307.356*	29.579	.000	229.269	385.443

Based on estimated marginal means

a. Adjustment for multiple comparisons: Bonferroni.

*. The mean difference is significant at the .05 level.

Univariate Tests

Dependent Variable:Completion_time

	Sum of Squares	df	Mean Square	F	Sig.
Contrast	4.948E7	3	1.649E7	49.521	.000
Error	1.018E9	3055	333063.232		

The F tests the effect of Menu_name. This test is based on the linearly independent pairwise comparisons among the estimated marginal means.

Page 3

Summary of Univariate ANOVA for touch pad device and select task, in experiment 5.

Univariate Analysis of Variance, touch, search, ratio

Between-Subjects Factors

		N
Menu_name	AAMUS	791
	AAMUS_Click	807
	AAMUS_Curvy	796
	Default	787
Ratio	60	1924
	75	1257
Participent No	18	162
	19	163
	20	164
	21	158
	22	164
	23	163
	24	161
	25	163
	26	157
	27	162
	28	161
	29	156
	30	153
	31	151
	32	160
	33	153
	35	164
	36	152
	37	156
	40	158

Tests of Between-Subjects Effects

Dependent Variable:Completion_time

Source		Type III Sum of Squares	df	Mean Square	F	Sig.
Intercept	Hypothesis	2.087E10	1	2.087E10	2766.938	.000
	Error	1.434E8	19.001	7.544E6		
Menu_name	Hypothesis	6.879E7	3	2.293E7	44.272	.000
	Error	2.957E7	57.087	517954.143 [b]		

a. .999 MS(ParticipentNo) + .001 MS(Error)

b. .999 MS(Menu_name * ParticipentNo) + .001 MS(Error)

c. .999 MS(Ratio * ParticipentNo) + .001 MS(Error)

d. 1.000 MS(Menu_name * ParticipentNo) + MS(Ratio * ParticipentNo) - 1.000 MS(Menu_name * Ratio * ParticipentNo)

e. .999 MS(Menu_name * Ratio * ParticipentNo) + .001 MS(Error)

f. MS(Menu_name * Ratio * ParticipentNo)

g. 1.000 MS(Menu_name * Ratio * ParticipentNo) + .000 MS(Error)

h. MS(Error)

Tests of Between-Subjects Effects

Dependent Variable:Completion_time

Source		Type III Sum of Squares	df	Mean Square	F	Sig.
Ratio	Hypothesis	9701350.119	1	9701350.119	10.588	.004
	Error	1.742E7	19.010	916300.700[c]		
ParticipentNo	Hypothesis	1.434E8	19	7547884.891	10.224	.000
	Error	7007438.280	9.492	738244.070[d]		
Menu_name * Ratio	Hypothesis	2306855.837	3	768951.946	1.105	.355
	Error	3.973E7	57.065	696163.085[e]		
Menu_name * ParticipentNo	Hypothesis	2.953E7	57	518008.779	.744	.867
	Error	3.969E7	57	696372.642[f]		
Ratio * ParticipentNo	Hypothesis	1.741E7	19	916535.620	1.316	.210
	Error	3.971E7	57.030	696274.827[g]		
Menu_name * Ratio * ParticipentNo	Hypothesis	3.969E7	57	696372.642	1.530	.007
	Error	1.375E9	3021	455105.306[h]		

a. .999 MS(ParticipentNo) + .001 MS(Error)

b. .999 MS(Menu_name * ParticipentNo) + .001 MS(Error)

c. .999 MS(Ratio * ParticipentNo) + .001 MS(Error)

d. 1.000 MS(Menu_name * ParticipentNo) + MS(Ratio * ParticipentNo) - 1.000 MS(Menu_name * Ratio * ParticipentNo)

e. .999 MS(Menu_name * Ratio * ParticipentNo) + .001 MS(Error)

f. MS(Menu_name * Ratio * ParticipentNo)

g. 1.000 MS(Menu_name * Ratio * ParticipentNo) + .000 MS(Error)

h. MS(Error)

Expected Mean Squares[a,b]

Source	Variance Component					
	Var (Participent No)	Var (Menu_name * ParticipentNo)	Var(Ratio * ParticipentNo)	Var (Menu_name * Ratio * ParticipentNo)	Var(Error)	Quadratic Term
Intercept	151.756	37.939	75.878	18.970	1.000	Intercept, Menu_name, Ratio, Menu_name * Ratio
Menu_name	.000	37.941	.000	18.970	1.000	Menu_name, Menu_name * Ratio
Ratio	.000	.000	75.878	18.970	1.000	Ratio, Menu_name * Ratio
ParticipentNo	151.834	37.958	75.917	18.979	1.000	
Menu_name * Ratio	.000	.000	.000	18.970	1.000	Menu_name * Ratio
Menu_name * ParticipentNo	.000	37.974	.000	18.987	1.000	
Ratio * ParticipentNo	.000	.000	75.917	18.979	1.000	
Menu_name * Ratio * ParticipentNo	.000	.000	.000	18.987	1.000	
Error	.000	.000	.000	.000	1.000	

a. For each source, the expected mean square equals the sum of the coefficients in the cells times the variance components, plus a quadratic term involving effects in the Quadratic Term cell.

b. Expected Mean Squares are based on the Type III Sums of Squares.

Estimated Marginal Means

2. Menu_name

Estimates

Dependent Variable:Completion_time

Menu_name	Mean	Std. Error	95% Confidence Interval	
			Lower Bound	Upper Bound
AAMUS	2567.452	24.539	2519.337	2615.567
AAMUS_Click	2495.930	24.320	2448.245	2543.615
AAMUS_Curvy	2546.921	24.518	2498.847	2594.994
Default	2879.808	24.585	2831.603	2928.013

Pairwise Comparisons

Dependent Variable:Completion_time

(I) Menu_name	(J) Menu_name	Mean Difference (I-J)	Std. Error	Sig.[a]	95% Confidence Interval for Difference[a]	
					Lower Bound	Upper Bound
AAMUS	AAMUS_Click	71.522	34.549	.231	-19.686	162.730
	AAMUS_Curvy	20.532	34.688	1.000	-71.046	112.109
	Default	-312.356*	34.736	.000	-404.058	-220.654
AAMUS_Click	AAMUS	-71.522	34.549	.231	-162.730	19.686
	AAMUS_Curvy	-50.991	34.533	.839	-142.159	40.178
	Default	-383.878*	34.581	.000	-475.172	-292.584
AAMUS_Curvy	AAMUS	-20.532	34.688	1.000	-112.109	71.046
	AAMUS_Click	50.991	34.533	.839	-40.178	142.159
	Default	-332.888*	34.721	.000	-424.550	-241.225
Default	AAMUS	312.356*	34.736	.000	220.654	404.058
	AAMUS_Click	383.878*	34.581	.000	292.584	475.172
	AAMUS_Curvy	332.888*	34.721	.000	241.225	424.550

Based on estimated marginal means

a. Adjustment for multiple comparisons: Bonferroni.

*. The mean difference is significant at the .05 level.

Univariate Tests

Dependent Variable:Completion_time

	Sum of Squares	df	Mean Square	F	Sig.
Contrast	6.879E7	3	2.293E7	50.386	.000
Error	1.375E9	3021	455105.306		

The F tests the effect of Menu_name. This test is based on the linearly independent pairwise comparisons among the estimated marginal means.

Summary of Univariate ANOVA for touch pad device and search task, in experiment 5.

BIBLIOGRAPHY

[1] Johnny Accot and Shumin Zhai. Beyond Fitts' law: Models for trajectory-based hci tasks. In *CHI '97: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 295–302, Atlanta, Georgia, United States, April 18–23 1997. ACM Press. ISBN 0-89791-802-9. doi: http://doi.acm.org/10.1145/258549.258760. (Cited on page 5.)

[2] Johnny Accot and Shumin Zhai. Performance evaluation of input devices in trajectory-based tasks: An application of the steering law. In *CHI '99: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 466–472, Pittsburgh, Pennsylvania, United States, May 15–20 1999. ACM Press. ISBN 0-201-48559-1. doi: http://doi.acm.org/10.1145/302979.303133. (Cited on pages 2, 6, and 18.)

[3] David Ahlstrom. Modeling and improving selection in cascading pull-down menus using Fitts' law, the steering law and force fields. In *CHI '05: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 61–70, Portland, Oregon, USA, April 2–7 2005. ACM Press. ISBN 1-58113-998-5. doi: http://doi.acm.org/10.1145/1054972.1054982. (Cited on pages viii, 5, 12, 13, 26, 33, and 79.)

[4] Benjamin B. Bederson. Fisheye menus. In *UIST '00: Proceedings of the 13th Annual ACM Symposium on User Interface Software and Technology*, pages 217–225, San Diego, California, United States, November 6–8 2000. ACM Press. ISBN 1-58113-212-3. doi: http://doi.acm.org/10.1145/354401.354782. (Cited on page 15.)

[5] J. Callahan, D. Hopkins, M. Weiser, and B. Shneiderman. An empirical comparison of pie vs. linear menus. In *CHI '88: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 95–100, Washington, D.C., United States, June 15–19 1988. ACM Press. ISBN 0-201-14237-6. doi: http://doi.acm.org/10.1145/57167.57182. (Cited on pages 5 and 8.)

[6] Andy Cockburn and Andrew Gin. Faster cascading menu selections with enlarged activation areas. In *GI '06: Proceedings of the 2006 Conference on Graphics Interface*, pages 65–71, Quebec,

Canada, June 7–9 2006. Canadian Information Processing Society. ISBN 1-56881-308-2. (Cited on pages viii, 2, 13, 14, 20, 26, and 79.)

[7] Leah Findlater and Joanna McGrenere. A comparison of static, adaptive, and adaptable menus. In *CHI '04: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 89–96, Vienna, Austria, April 24–29 2004. ACM Press. ISBN 1-58113-702-8. doi: http://doi.acm.org/10.1145/985692.985704. (Cited on page 9.)

[8] Paul Fitts. The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology*, 47:381–391, 1954. (Cited on page 4.)

[9] Tovi Grossman and Ravin Balakrishnan. The bubble cursor: Enhancing target acquisition by dynamic resizing of the cursor's activation area. In *CHI '05: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 281–290, Portland, Oregon, USA, April 2–7 2005. ACM Press. ISBN 1-58113-998-5. doi: http://doi.acm.org/10.1145/1054972.1055012. (Cited on pages 5 and 15.)

[10] Masatomo Kobayashi and Takeo Igarashi. Considering the direction of cursor movement for efficient traversal of cascading menus. In *UIST '03: Proceedings of the 16th Annual ACM Symposium on User Interface Software and Technology*, pages 91–94, Vancouver, Canada, November 2–5 2003. ACM Press. ISBN 1-58113-636-6. doi: http://doi.acm.org/10.1145/964696.964706. (Cited on pages viii, 1, 10, 11, 19, 26, 33, and 79.)

[11] Gordon Kurtenbach and William Buxton. User learning and performance with marking menus. In *CHI '94: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 258–264, Boston, Massachusetts, United States, April 24–28 1994. ACM Press. ISBN 0-89791-650-6. doi: http://doi.acm.org/10.1145/191666.191759. (Cited on page 8.)

[12] I. Scott MacKenzie. Fitts' law as a research and design tool in human-computer interaction. *Human-Computer Interaction*, 7: 91–139, 1992. (Cited on pages 4 and 5.)

[13] I. Scott MacKenzie. Movement time prediction in human-computer interfaces. In *Conference on Graphics Interface*, pages 140–150, Vancouver, British Columbia, Canada, 1992. Morgan Kaufmann Publishers Inc. ISBN 0-9695338-1-0. (Cited on page 4.)

[14] I. Scott MacKenzie and William Buxton. Extending Fitts' law to two-dimensional tasks. In *CHI '92: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 219–226, Monterey, California, United States, May 3–7 1992. ACM Press. ISBN 0-89791-513-5. doi: http://doi.acm.org/10.1145/142750.142794. (Cited on page 5.)

[15] Peter Oel, Paul Schmidt, and Alfred Schmitt. Time prediction of mouse-based cursor movements. In *IHM-HCI'2001: Conference on Human-Computer Interaction*, pages 37–40, Lille, France, September 10–14 2001. (Cited on page 5.)

[16] Robert Pastel. Measuring the difficulty of steering through corners. In *CHI '06: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1087–1096, Montreal, Quebec, Canada, April 24–27 2006. ACM Press. ISBN 1-59593-372-7. doi: http://doi.acm.org/10.1145/1124772.1124934. (Cited on pages 2, 6, and 19.)

[17] Andrew Sears and Ben Shneiderman. Split menus: Effectively using selection frequency to organize menus. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 1(1):27–51, 1994. ISSN 1073-0516. doi: http://doi.acm.org/10.1145/174630.174632. (Cited on page 8.)

[18] Erum Tanvir, Jonathan Cullen, Pourang Irani, and Andy Cockburn. AAMU: Adaptive activation area menus for improving selection in cascading pull-down menus. In *CHI '08: Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, pages 1381–1384, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-011-1. doi: http://doi.acm.org/10.1145/1357054.1357270. (Cited on page 26.)