

AAMU: Adaptive Activation Area Menus for Improving Selection in Cascading Pull-Down Menus

Erum Tanvir
University of Manitoba,
Winnipeg, Canada.
etanvir@cs.umanitoba.ca

Jonathan Cullen
University of Manitoba,
Winnipeg, Canada.
umculle3@cc.umanitoba.ca

Pourang Irani
University of Manitoba,
Winnipeg, Canada.
irani@cs.umanitoba.ca

Andy Cockburn
University of Canterbury,
Christchurch, New Zealand.
andy@cosc.canterbury.ac.nz

ABSTRACT

Selecting items in cascading pull-down menus is a frequent task in most GUIs. These selections involve two major components: steering and selection, with the steering component being the most time-consuming and error-prone. We describe a new technique, called Adaptive Activation-Area Menu (AAMU) that eliminates corner steering. AAMUs contain an enlarged activation area which dynamically resizes itself providing a broader steering path for menu navigation. We also combined AAMUs with Force-field menus, to create Force-AAMUs. We empirically demonstrate that AAMUs and Force-AAMUs outperformed the current default menu. We also compared performances of various other menus including Enlarged activation area menus (EMUs) and Gesture based selection with mouse as an input device. Overall, users show higher satisfaction rates for AAMUs over other menu designs.

Author Keywords

Cascading pull-down menus, menu navigation, steering, selection, Force field menus, AAMU, Force-AAMU.

ACM Classification Keywords

H5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous.

INTRODUCTION

Menus are an important part of a graphical user interface (GUI), providing users a convenient and familiar means of command selection. Cascading submenus are frequently used to avoid over-long menu lengths, and to group related items into a hierarchical structure. However, they have a substantial limitation such that users must carefully steer their cursor through a narrow selection pathway, as demonstrated in Figure 1. Deviating from the correct path can cause unexpected selections and unintended invocation of submenu appearance or disappearance.

To reduce the impact of steering errors, traditional

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

CHI 2008, April 5–10, 2008, Florence, Italy.

Copyright 2008 ACM 978-1-60558-011-1/08/04...\$5.00

cascading menus introduce a time delay between entering a parent menu item and displaying the cascading menu. To speed up the navigation through the hierarchy, users can pre-empt the delay with explicit clicks, which transforms the fluid but delayed steering activity into a series of discrete target acquisitions. Calibrating the duration of the delay to maintain fluid and rapid steering behavior without the distraction of steering errors is complex because preferences are likely to vary between users, devices and contexts of use such as desktop/mobile use. [6].

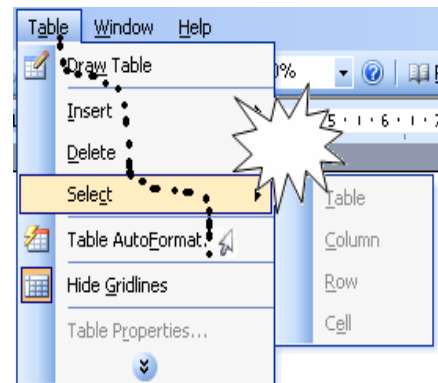


Figure 1 Steering errors in cascade menu selections can cause submenus to appear and disappear.

Several researchers have investigated methods for improving item selection in cascading drop-down menus. Examples include rearranging menu items [6], gesture based menus [4], enlarged activation areas [2] and force enhanced menus [1]. However, all of these techniques have their limitations. In this paper, we introduce a novel technique called *Adaptive Activation Area Menu* or *AAMU* based on the premise that users can be more efficient without steering through sharp corners [5]. The steering task is facilitated by providing the user with an adaptive activation area which dynamically resizes itself according to the size of the child cascading menu. This provides the user a broad path to steer and allows diagonal movements. The study compared AAMUs to a number of existing techniques including Enlarged activation area menus (EMUs) [2], Gesture based selection [4], Force fields [1] and Force-AAMUs. The results of the study show that users can efficiently perform menu selection when provided with a broader steering path.

The main contribution of this paper is the introduction of a new cascading menu technique, AAMU, and its variant Force-AAMU. We demonstrate that we can improve performance with cascaded menus by avoiding steering through sharp corners.

SELECTING ITEMS FROM CASCADING MENUS

Cascading menu is the most commonly used technique for handling hierarchical menus. However, selecting items from cascading menus is prone to errors. Additionally, conventional cascading menus are implemented with an explicit delay for the posting/un-posting of the child submenu which makes the selection process very slow. With the increase in complexity and size of cascading menus, there is a growing demand for improving their design in order to make the navigation and selection process faster and easier. In this section we review various existing improvements to the selection of items in cascading pull-down menus.

Researchers have designed different menu organization schemes for pull-down menus to reduce targeting requirements and to speed up the performance. Split menus [6] and Adaptive split menus [3] are examples that have appeared in commercial and common everyday environments. However, Split and Adaptive menus still are challenged by the need for steering around sharp corners.

Kobayashi and Igarashi [4] presented an improvement to cascading menus by reducing the steering distance and avoiding the unintended menu posting/un-posting. In this technique the horizontal motion of the mouse cursor opens/closes the child submenus, eliminating the unwanted submenu appearance during menu navigation. Upon invocation, submenus pop up beneath the cursor position hence reducing the length of the movement path. However this technique is not widely applicable since it adds additional movements to open/close submenus which can slow down the interaction process and cause fatigue in users. Furthermore, in this technique child submenus overlap their parent menus and hide the rest of the parent menu items. The resulting overlap forces users to make additional horizontal movements to close a submenu before they could interact with the parent menu.

Another technique developed by Cockburn and Gin [2] is called Enlarged activation-area menus or EMUs. EMUs improved navigation through cascading menus by increasing the activation area associated with each cascaded submenu, thereby providing a wider path for steering. Also, EMUs allow faster selection performance by eliminating the problem of time delay in posting/un-posting submenus. A major limitation of EMU is that the activation area is enlarged depending on the density of the cascaded items in the parent menu. As a result, in the worst case scenario, where all parent items have submenus, the size of the activation area will be equal to that of the traditional cascading menu hence offering no performance benefits.

To make the selection process faster, Ahlstrom [1] introduced “force fields” in traditional cascading pull down menus. Force field menus partially took control of the cursor movement from the users. When moving from left to right, within a parent menu item, the cursor was “pushed” towards the child menu and moved faster, thus optimizing the selection process. The most significant benefit of force field menus was that it kept the visual structure of the interface and the interaction technique unchanged.

ADAPTIVE ACTIVATION AREA MENUS (AAMUs)

In this paper we introduce a new menu type, the Adaptive Activation Area Menu (AAMU). The aim of this technique is to reduce selection/movement errors by eliminating the need to steer through sharp corners and without forcing the users to wait (Figure 2). AAMUs provide the additional benefit in steering without changing the traditional structure and the interaction technique of the cascading menus. This is achieved by providing users with a triangle shaped enlarged activation area, allowing them to steer diagonally as well as horizontally (Figure 2b).

With AAMUs, when the cursor enters a parent menu item that has a submenu, the parent item is highlighted and a triangular activation area appears near the cursors' current position. The other two vertices of the triangle connect to the top and bottom left corners of the child menu. As a result, the size of the activation area is determined dynamically depending on the size of the child submenu and the cursors' position. The activation area is transparent and therefore it does not occlude the rest of the parent menu items. This also allows the users to view and select other parent items without explicitly closing the activation area. The activation area lies on top of the parent menu and will remain open as long as the cursor remains in the confines of the cascaded item or within the activation area. If the cursor enters over any other menu item the activation area disappears. If the cursor enters into a parent with a cascaded item, a new activation area will reappear.

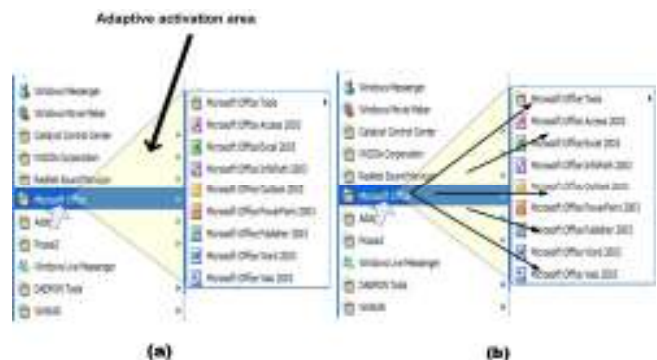


Figure2: (a) Adaptive activation area. (b) The possibilities for broader steering.

We hypothesized that a broader steering path will reduce selection errors and make the navigation process easier. Additionally the dynamic activation area eliminates the

need for imposing a time delay while posting/un-posting a child menu, hence further reducing selection time.

We also designed a variant of AAMU that combines AAMU with force fields, naming them Force-AAMUs. While providing the benefits of diagonal navigation, Force-AAMUs further speed up the selection and navigation process by increasing the speed of the cursor movement within the activation area. We implemented the force fields in the menu item as well as within the activation area. Unlike the original force field menus, in the Force-AAMU, force fields are only applied in the forward motion and there is no resistance in backward direction.

STUDY OF AAMU

The study compared AAMUs to a number of existing techniques including Enlarged activation area menus (EMUs) [2], Gesture based selection [4], Force fields [1] and a new combination technique called Force-AAMUs. For the user study, all menu types were implemented without any time delay. We compared AAMUs against the strongest existing cascading menu designs to provide a fair perspective into the merits of each technique. Also, we wanted to show where each technique stands, with respect to performance and user preference.

Materials

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

Methods

Goals

The goal of this study was to analyze the performance of various menu designs while selecting menu items at various levels of menu depths. All the menus were designed without using the time delay. We hypothesized that AAMUs and Force-AAMUs will further reduce the selection time by alleviating the steering problem.

Participants

Eleven university undergraduate students participated for course credit. All of them had used the MS Windows default menu and were familiar with operating a mouse.

Task and Stimuli

Participants were required to perform 30 menu selection tasks with each interface, with 10 trials at each of three cascading menu depths (2, 3, and 4). A path to a particular menu item was provided by highlighting it in green color. The target menu item was displayed in red color. 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 level of menu depth. The positioning of the target item was determined randomly. For each trial a different path and

target was 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. The experiment took approximately 25 minutes per participant.

Design

The logged dependent variable (task time) was analyzed using a 6×3 repeated measures analysis of variance 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

There was a significant main effect of *interface type* ($F_{5,50}=28.5$, $p<.001$), with AAMUS performing fastest (mean 1.93s, sd 0.55), followed by force-AAMUS (1.96s, 0.58), force fields (2.05s, 0.62), EMUs (2.28s, 0.75), default (2.53s, 0.91) and gesture (3.13s, 1.36). As expected there was a significant main effect for *depth* ($F_{2,20}=172.4$, $p<.001$). There was also a significant *interface×depth* interaction ($F_{10,100}=8.9$, $p<.001$). These results are summarized in (Figure 3). 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.

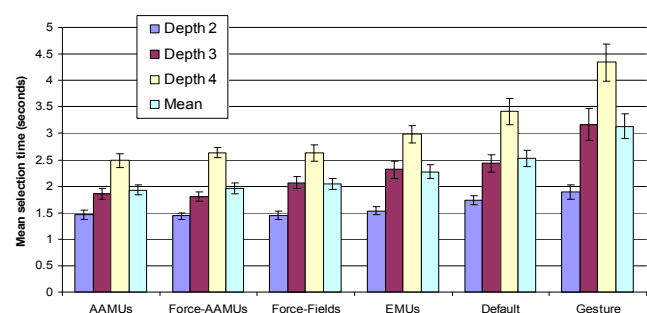


Figure3: Selection times for various Menu types

A Tukey post-hoc comparison of the difference between interface types gives an honest significant difference of 0.63s ($p<.05$), indicating significant differences between the gesture interface and all others, and a marginal difference between the default method and AAMUs.

Subjective Rankings

A post-study questionnaire was collected from participants asking for their most preferred technique. 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 (Figure 4). Users gave lower preference to EMUs due 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 speed 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.

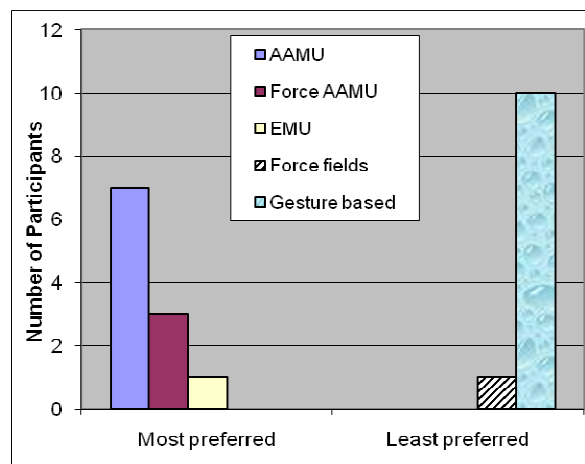


Figure 4: Number of participants that ranked the techniques as either most preferred or least preferred.

DISCUSSION

Our results show a marginal improvement in performances with AAMUs over the Default cascading menu style. AAMUs improved selection times by 23.5% when compared to Default menus, by 15% against EMUs and 5.6% against Force fields. Force-AAMU improved the selection times by 22% when compared to default menus, by 14% as compared to EMUs and 4% against Force fields. We also logged the total number of clicks to calculate the selection errors caused while using various menu types. For depth level 2, AAMUs, Force-AAMUs and Default showed no selection errors where as Force fields, EMU and gesture based showed errors for all participants. Overall, AAMUs reduced the selection errors by 4.7% against Default, 4.4% against EMU and 2% against Force fields.

Overall, users preferred AAMUs and Force-AAMUs due to the ease of steering in selection tasks. These techniques also provided more control to the user with the cursor. Users particularly disliked the Force enhanced menus due to loss of control over menu navigation and this contributed to higher error rates in Force fields and Force-AAMUs.

One limitation of AAMUs is that if the activation area is fully expanded then the user could not move to the adjacent menu item immediately. We call it the “cursor trapping” problem and the user needs to get out of the activation area before entering the next menu item. However, as the adaptive activation area starts right at the cursor position, therefore the user need not alter the navigation path to a great extent and this problem does not affect the overall menu navigation and selection process.

CONCLUSION

We presented Adaptive Activation Area Menu that was designed to reduce errors caused due to steering through sharp corners. AAMU provides users with a broader pathway while selecting items from pull-down cascading menus. Additionally, AAMUs eliminate the time delay for posting/un-posting submenus. We also combined AAMUs with force fields to create a variant technique (Force-AAMUs) with fast cursor movements. The evaluation showed that AAMUs and Force-AAMUs improved the selection performance by 22% in comparison to the Default submenu invocation mechanism. Users also expressed higher satisfaction for AAMUs. The only limitation of AAMUs is that it introduces diagonal steering and users currently are more familiar with steering through corners. However, with practice, we expect users will become comfortable with diagonal steering. In future work, we are evaluating AAMUs with other types of tasks. We are also designing interface widgets based on AAMUs and will be evaluating their performance in current day-to-day menu navigation environments, such as with web interfaces.

ACKNOWLEDGEMENTS

We thank NSERC for funding part of this research, the reviewers for insightful comments, and the participants for their valuable time and effort.

REFERENCES

- Ahlstrom, D. Modeling and improving selection in cascading pull-down menus using fitts' law, the steering law and force fields In Proc. CHI '05, 61–70, 2006.
- Cockburn, A. and Gin, A. Faster cascading menu selections with enlarged activation areas. In Proc. GI'06, 65–71, 2006.
- Findlater, L. and McGrenere, J. A comparison of static, adaptive, and adaptable menus. In Proc. CHI '04, 89–96, 2004.
- Kobayashi, M. and Igarashi, T. Considering the direction of cursor movement for efficient traversal of cascading menus. In Proc. UIST '03, 91–94, 2003.
- Pastel, R. Measuring the difficulty of steering through corners. In Proc. CHI '06, 1087–1096, 2006.
- Sears, A. and Shneiderman, B. Split menus: Effectively using selection frequency to organize menus. ACM Transactions on Computer- Human Interaction (TOCHI), 1(1):27–51, 1994.