Simple Mark Hierarchical Marking Menus

Project Paper

Svetlana Slavova
sds797@mail.usask.ca
Department of Computer Science
University of Saskatchewan

ABSTRACT
This document is a project paper about simple mark hierarchical marking menus. It discusses why these menus are important, as well as what advantages they offer, compared to some other types of menus (such as linear and pie). In addition, the paper represents two menu designs, their implementation and evaluation. The results of the experiments show that action backing-up is an important feature which increases the selection of multi-level items. Finally, the paper presents some visual design improvements which help create a more compact and easy to use simple mark hierarchical marking menu.

Keywords
Marking menus, Simple Mark Hierarchical Marking Menus, Graphical User Interface.

INTRODUCTION
Menus are part of the user interface of every software system [1]. They are applied in various devices, such as personal computers, laptops, PDAs, BlackBerries, and cell phones. The difference and the limitations of the input (keyboard, mouse, pen, etc.) and the output (display size, screen resolution, etc.) of the ubiquitous computing devices require appropriate menus to be applied, in order to provide appropriate interface to the user. There are several styles of menus, considered in the literature – linear, pie (radial), and marking menus ([1], [2], [3], [4], and [5]). The linear menus, such as pop-up and pull-down menus, list items from top to bottom of the window, whereas the pie menus display the icons in the form of a circle. The "marking menus are a refinement of radial (or pie) menus" [3]. The marking menus, as well as the other two styles, can be organized in hierarchies, according to the categories of the items. The marking menus allow the expert users to work more efficiently, since instead of showing the menu, a simple mark or a set of marks can be performed, in order to select a particular item. Different studies [2] show that the marking menus are faster, more reliable, and allow smooth transition from a novice to an expert level of usage, compared to the other menu styles.

PROBLEM DEFINITION
When operating with a hierarchical marking menu, users perform compound zigzag actions, in order to select an item from a multi-level menu [2]. As a result, the number of errors of the selection process has a significant value, "as the number of levels (menu depth) and items per level (menu breadth) in the menu hierarchy increases". This is caused by the length of the mark that has to be made, as well as by the close similarity of some marks, which refer to different items. Furthermore, it is even possible that the display size of the device does not allow a long compound mark to be drawn.

All these problems lead to the following questions:

- Is it possible to select a multi-level menu item with a set of simple marks, instead of a compound one?
- If yes, then how to represent a simple mark hierarchical menu?

RELATED WORK
Marking menus have the potential to be used along with or even as an alternative of the linear menus [1]. The main reason for this result is that the marking menus provide a faster selection with a reduced error rate, especially when the layout of the menu is known by the user [1], [3]. In addition, the ease tradition from a novice to an expert level of usage is another big advantage of the marking menus, which increases the selection speed.

However, there is a limitation regarding the number of items per sub-menu (breadth) and number of sub-menus (depth) in the marking menu. Compound marking menus, constructed as 4 items per level with depth of 4 levels (4x4), as well as 8 items per level with depth of 2 levels (8x2), allow fast selection with low error rate [5]. However, the number of errors increases when breadth and/or depth of the menu increases.

Tapia and Kurtenbach [3] propose refinements of marking menus that improve the menu design regarding "maintaining visual context", "hiding unnecessary information", and "supporting skill development by graphical feedback". As a result they achieve a better

visualization of the menu, which requires less space due to better organization of the labels, less hand movement, as well as it provides useful feedback to the user, indicating how the application understands the completed action. In addition, the refinement menu hides the upper level, when an item is selected, which however makes it harder to perform backing-up.

Zhao and Balakrishnan [2] go even further and present a hierarchical marking menu, which is based on simple marks, rather than on compound zigzag actions. They compare the behavior of two menu designs – compound and simple mark hierarchical marking menus. The results show that the simple marks menu allows increasing the number of items in the menu, because it does not contain ambiguous marks. Since every action is uniquely defined by a set of independent marks, overlapping submenus can be constructed, in order to reduce space for menu visualization. However, the authors have not investigated different cases regarding navigation through simple marking menu.

A mark can be uniquely identified by its length, position, and orientation [4]. Studies show that the length of the mark is hard to be remembered. Bull's eye is an example of a menu that takes into consideration the length of the mark. The items of the menu are placed in concentric circles and an item is selected depending on the length of the mark. In order to be effective, this menu type must provide continuous feedback to the user.

Zhao, Agrawala, and Hinckley [4] base their approaches on both the position and the orientation of the marks. They introduce zone and polygon hierarchical marking menus in order to augment the number of items per level in the menu. The zone menu consists of multiple zone areas. Depending on the menu, a zone might represent one or more items. The orientation of the mark uniquely identifies the desired item within a zone. The polygon menu is in the form of a polygon and the number of items depends on the number of sides of the polygon. "The breadth of an N-sided polygon menu is 2*N" [4]. As in the zone menu, the selected item is defined by the orientation and the position of the drawn mark. The results of the evaluation of these menu types show that the breadth of the menu can be increased to 16 items, while keeping high selection performance in terms of speed and accuracy.

This project paper is based on the idea of [2] to construct a simple mark hierarchical marking menu, but in addition, it compares two different navigation styles through the menu – with and without backing-up.

SIMPLE MARK HIERARCHICAL MARKING MENUS

Motivation

The simple mark hierarchical menus consist of a set of marks that has to be drawn, in order to select an item from the menu [2]. When using simple marks, each line is shorter than the zig-zag lines of compound marking menu. This indicates that the error rate of the items selection should decrease, due to the length of the mark.

As discussed in [2], the problem with the similarity of some compound marks should not longer exists, due to the fact that each set of marks is uniquely defined by the number of independent simple marks in it, as well as by their directions.

On the other hand, the independence of the simple marks allows the construction of overlapping sub-menus [2]. Thus, the display size of the device should not be a limitation any more for small devices that require visualization and manipulation of multi-level menus.

Design

A simple mark hierarchical menu consists of three main components:

- *Start position*. The start position of the menu is used to show the menu to a novice user, as well as it indicates where the beginning of the marking should start;
- *Items*. The items form each level of the hierarchical marking menu. There are two types of items – atomic and compound. The atomic item is a selectable item, which provides a particular command, whereas the compound one consists of items, representing a sub-level in the menu. The colors of the atomic and the compound item are different, so that the user is able to make a distinction between them. Once a particular atomic item is selected, it is highlighted in a third color, indicating that the item is activated;
- *Simple mark*. The simple mark is used in order to select an item or a sub-menu. Depending on the device, the mark can be realized with different input devices, such as a pen or a mouse.

The actions that can be applied to the hierarchical marking menu are as follows:

- *Draw a mark*. The type of input device defines how the mark to be drawn. In case of a pen, the steps are the following: (1a) the pen is located to the start position; (2a) the pen starts drawing a mark in direction towards the item that is to be selected; (3a) when the pen has moved from the start position to outside of the end boundary of the item, the item is considered as selected. Similarly, the mark can be drawn by a mouse, as follows: (1b) the mouse is located to the start position of the menu; (2b) the event from action left button-down indicates the beginning of the marking; in addition, the mouse should start moving towards the desired item; (3b) when the mouse cursor is outside of the end boundary of the item and the left button-up event occurs, the item is considered as selected.
- *Cancel action for mark drawing*. The user simply needs to lift up the pen / release the mouse button. This action cancels the last attempt of drawing a mark.

- *Cancel action for item selection*. In order to cancel the whole selection, the user should make a short mark starting outside of the start position of the menu.
- *Backing-up*. Backing-up is an action that causes the last selected item (if any) to be ignored. I.e. the current level of the menu becomes the upper level. In order to do this action, a single dot in the start position must be drawn – in case of a pen: releasing the pen on the start position and lifting it up, whereas in case of a mouse: events left button-down and left-button up (i.e. mouse click) on the start position of the menu must occur.

For this project, the start position is located in the middle of the menu; the number of items per menu-level is four; and the marks are realized by a mouse.

Two different types of single mark hierarchical marking menus are designed. The first one allows the actions – draw a mark, cancel action for mark drawing, and cancel action for item selection, whereas the second one allows the actions of the first design, including backing-up. Thus, the experiments will show whether or not the backing-up feature is useful; or to start the item selection from the beginning is a better approach.

IMPLEMENTATION

An application, which represents simple mark hierarchical marking menus without and with backing-up, is developed in Java, using Java Swing. Figure 1 and figure 2 show the main window of the program as well as the design of the simple mark hierarchical marking menu. Figure 3 shows an example how to perform the menu actions. In addition to the described actions that can be applied to the menus, the application is able to navigate the user through the menu using sound effects. However, the sound-effects are not used during the evaluation process since the participants are willing to wait until hear the corresponding sound of the selected item.

Figure 1. Main window of the application

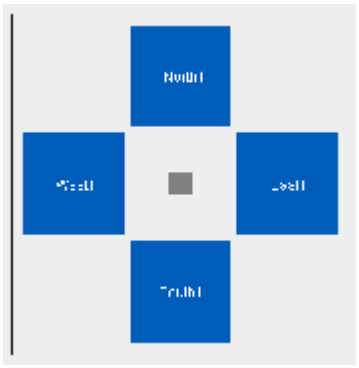

Figure 2. Simple mark hierarchical marking menu

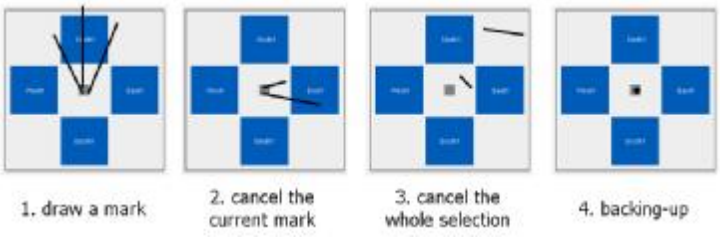

Figure 3. Actions of the simple mark hierarchical marking menu

Table 1. Implemented classes in Java

N	Class name	Description
1	**RunMenu**	The starting class of the application. It shows the main window of the program.
2	**MenuFrame**	It represents a window with a menu.
3	**MenuPanel**	It is responsible for the graphical visualization of the menu.
4	**HierarchicalMenu**	It represents the menu as a collection of items, which are located on different levels of the menu.
5	**MenuItem**	It represents an item of the menu. An item could be atomic or a compound item which contains other items.
6	**Experiment**	It contains functionality, necessary for the experiments. In addition, it writes the results in a log file.

Steps of the implementation

1. Planning of the functionality and the GUI of the application;
2. Paper prototype of the GUI of the desired application;
3. Implementation of a simple mark hierarchical marking menu without backing-up;
4. Implementation of a simple mark hierarchical marking menu with backing-up;
5. Implementation of the experiments of the application.

The developed classes are described in table 1 and their hierarchy is presented in figure 4.

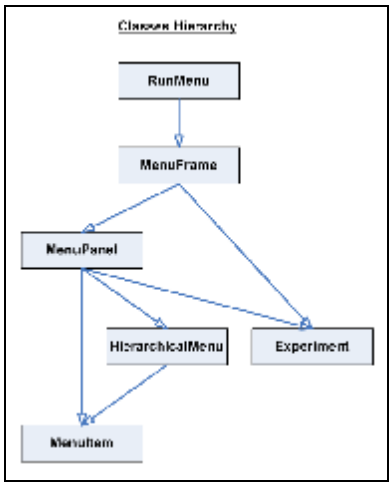

Figure 4. Hierarchy of the implemented classes

EVALUATION

Goal

The goal of the evaluation is to show whether or not the simple marks are an applicable technique in multi-level menus. In addition, two design models are compared, in order to analyze and suggest an appropriate set of functionality that a marking menu should provide.

Experiment

Every menu design checks the user's ability to navigate in depth through the hierarchical marking menu. The two proposed approaches are evaluated using the same set of start conditions (independent variables). This allows accurate comparison between the design models.

The independent variables of the experiment setup are:

- *Items & levels*. The design models consist of the same set and order of selectable items and menu levels – 4 items per level, 3 levels per menu design. A compound item is represented in blue, an atomic item is highlighted in dark grey, and a selected atomic item is colored in green;
- *Start position*. It is the same for each menu;
- *Users*. The same users test both models. At every step of the experiment, they are informed what action is performed/item is selected. After a trial completion, the application displays whether or not it is done successfully;
- *Paths*. The users are given paths through the menu, indicating which items should be selected. The paths are the same for both menu designs. Within a design, the paths specify items, located in different levels of the menu.

The dependent (observed) variables are the following:

- *Error rate*. The error rate defines if an item selection is accurate or not. Higher the error rate, lower the precision;
- *Time for item selection*. Shows how much time is needed to select a desired item from the menu;
- *Number of simple marks*. The number of marks indicates how many marks are needed in order to select an item.

Ten different paths are considered, representing five path types, as follows: 1-level return, 2-level returns, 2 1-level returns, and 2 2-level returns. When dealing with a simple marking menu without backing-up, the return is realized via action cancel the whole selection, whereas for simple marking menu with backing-up, the return is realized via one or more consecutive actions baking-up. Two paths per path type, requiring different number of marks to be made, are taken into account. Every path is repeated three times. The total number of trials per menu design is 30, i.e. 60 trials for the whole experiment per participant.

The subjects, who participated in the experiments, are graduate students at the Department of Computer Science. Six subjects were able to take part in the experiments. The experiments were conducted at my work place at MADMUC Lab at the Department of Computer Science. The subjects had the same conditions during the experiments – computer, mouse, monitor, and the simple mark hierarchical marking menu application. Before starting the experiments, the subjects were told about the possible actions they can perform in the application. In addition, they were able to practice all actions. The experiments started when the subjects were ready and after a small practice period. At the end of the experiments, the participants were able to give feedback regarding how they felt about the simple mark hierarchical marking menus during the experiment phase.

Figure 5 shows the experiment window for simple mark hierarchical marking menu without backing-up. The user is informed about the current error rate, the current number of trial, as well as the current and the destination path. When the next trial is started, the application displays the result of

the previous trial completion and updates the error rate. A trial is completed successfully when the current and the destination path are the equal. The same window appears for simple mark hierarchical marking menu with backing-up, but the destination path asks the user to perform action backing-up, instead of action cancel the whole selection.

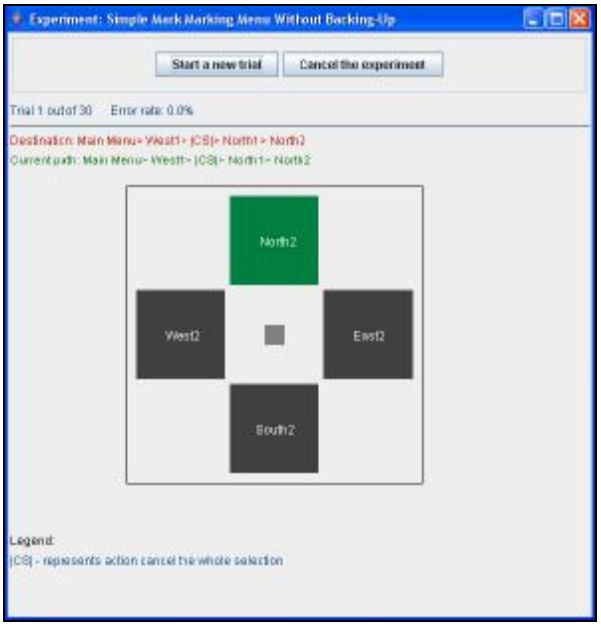

Figure 5. Experiment: Simple mark hierarchical marking menu without backing-up

Results

When the number of marks necessary for item selection increases, the time increases as well. This effect, presented in tables 2 and 3, is observed for both menu designs.

Figure 6 represents the time comparison between the two menu designs. The results show that although the menu with backing-up requires more marks to be drawn in order to perform a specific item selection, it is faster than the menu without backing-up. The reason for this result is that it is much easier to perform action backing-up than the action that cancels the whole selection: Action backing-up requires a simple dot in the start position of the menu to be made, whereas in order to cancel the whole selection, the user must draw a mark starting outside of the start position of the menu.

Figure 7 represents the error rate comparison between both menus. The menu with backing-up has a stable error rate than the other menu design. The reason for this result is that it is more intuitive to perform action backing-up than to cancel the whole selection and to start the selection from the beginning. I.e. while performing backing-up, the user navigates through the menu by levels, whereas when the whole selection is canceled, the user is more likely to loose orientation. However, backing-up is not convenient for repeated returns through the menu, because this increases the error rate, as shown for 2 2-level of returns in the menu with backing-up. In addition a task of such complexity is not likely to occur in real-world situations.

Table 2. Results without backing-up

	Marks	Time, ms	Error rate, %	Standard error, ms
Without return	2	1,713.5	0	101.2
	3	3,126.7	0	183.2
1-level return	4	5,258.8	11.1	647.5
	5	7,163.4	5.6	425.3
2-level returns	5	7,275.3	11.1	474.3
	6	8,607.2	16.7	548.1
2 1-level returns	6	9,536.8	5.6	907.3
	7	11,167.7	0	874.8
2 2-level returns	8	13,166.3	5.6	840.4
	9	15,099.3	5.6	970.3

Table 3. Results with backing-up

	Marks	Time, ms	Error rate, %	Standard error, ms
Without return	2	1,375.8	5.6	66.5
	3	2,775.1	0	124.4
1-level return	4	4,125.1	0	259.3
	5	5,799.8	5.6	429.2
2-level returns	6	6,295.9	0	232.6
	7	8,192.0	5.6	573.1
2 1-level returns	6	6,577.3	0	279.6
	7	8,014.8	0	452.9
2 2-level returns	10	10,396.8	16.7	532.5
	11	11,707.8	27.8	405.5

Figure 8 represents the error rate comparison between the participants involved in the evaluation process. The figure shows that participant #6 has more than 15% error rate for both menu designs. A comparison of the results obtained from 5 and 6 participants shows that there is not a difference in the interpretation of the obtained data. I.e. the same characteristics are observed in the diagrams with 5 and 6 participants and the same conclusions can be made. This means that although the last subject has a higher error rate, the gathered results do not change the interpretation of the observed dependent variables.

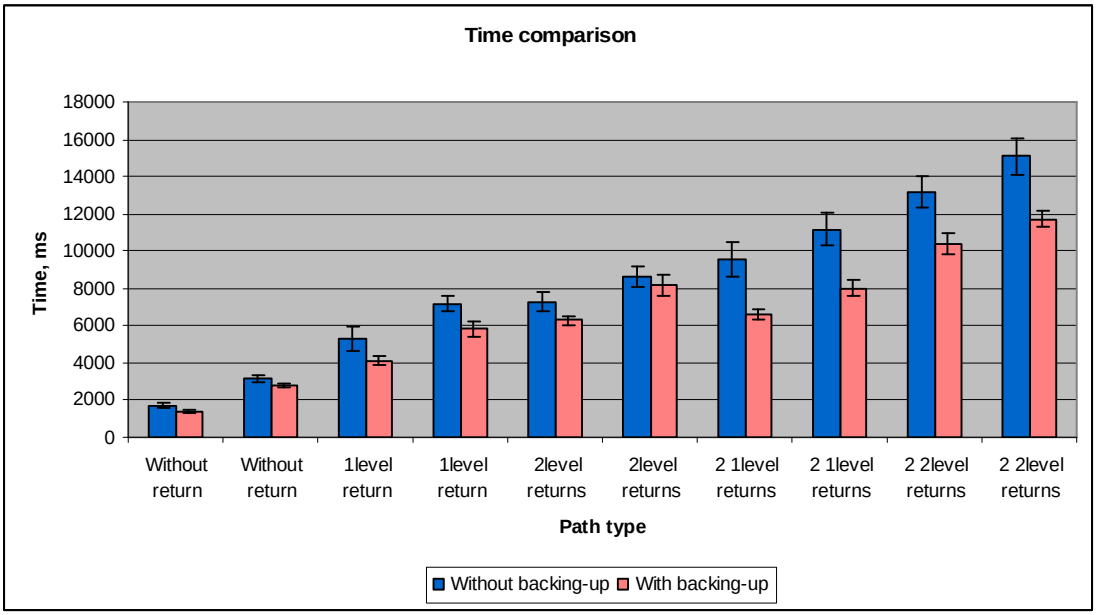

Figure 6. Time comparison

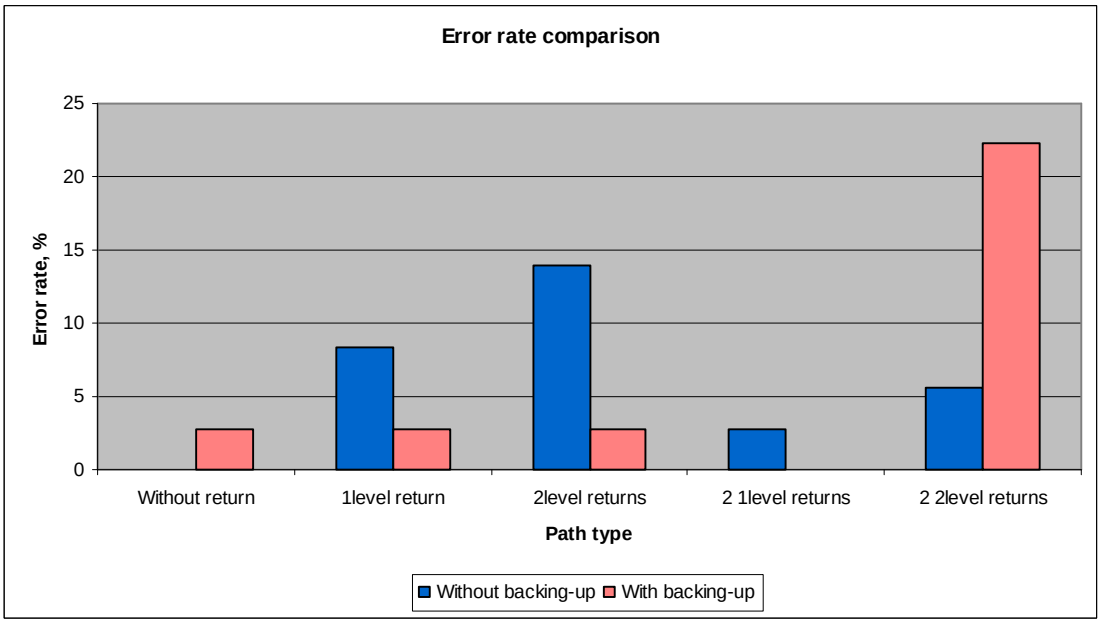

Figure 7. Error rate comparison

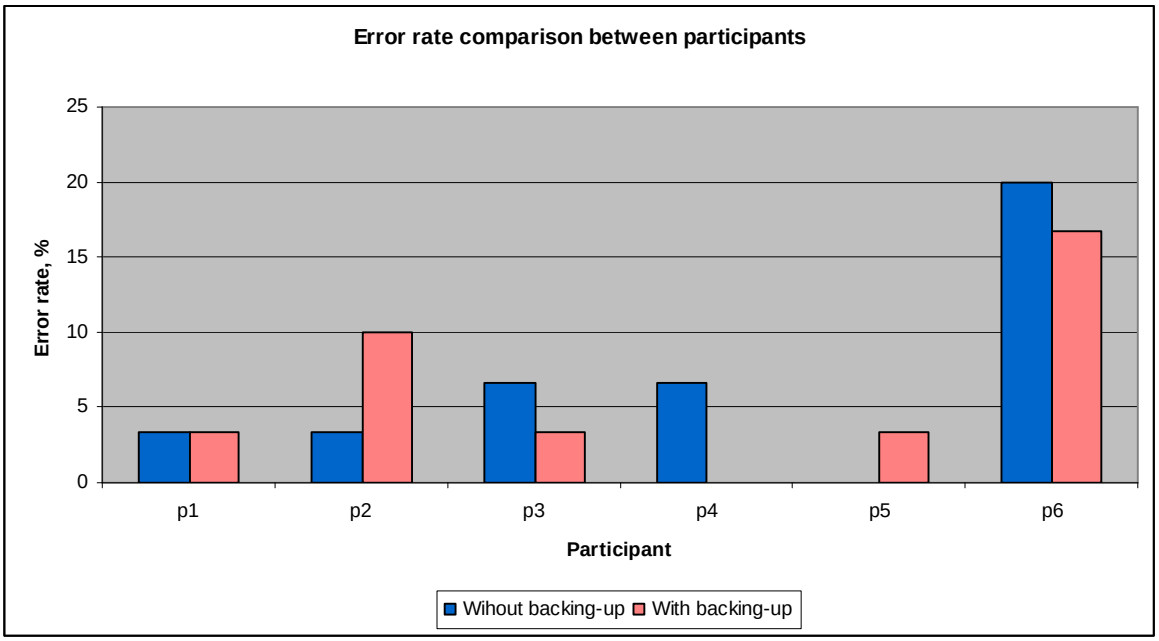

Figure 8. Error rate comparison between participants

DESIGN IMPLICATIONS

The menus, presented in this paper, allow multi-level item selection using simple marks. The participants' feedback collected during the evaluation period of the application suggests that improvements in the menu design must be done. The observations indicate that the start position of the marking menu should be enlarged, as shown in figure 9. This would allow an easier item selection, since the target item can be reached by a higher number of potential marks. In addition, the width of the items can be smaller, which requires a shorter mark to be drawn and allows a faster selection.

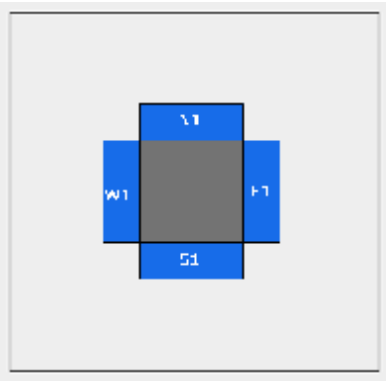

Figure 9. Improved design of simple mark hierarchical marking menu

CONCLUSIONS & FUTURE DIRECTIONS

Menus are widely used in software applications and are important part of the GUI of mobile and ubiquitous devices. Three main types of menus are presented in the literature – linear, pie, and marking menus.

This project focuses on simple mark hierarchical marking menus and discuses the following questions:

- Is it possible to select a multi-level menu item with a set of simple marks, instead of a zigzag mark?
- If yes, then how to represent a simple mark hierarchical marking menu?

The presented designs of a simple mark hierarchical marking menu show that they are a feasible technique and can be used for multi-level item selection. The nature of the simple marking menu allows constructing overlapping menus, which reduces the required space. This is useful for small screen devices providing limited working area. In addition, every set of marks is uniquely defined by the number of marks in it, as well as by their direction. This suggests that there are no similar marks corresponding to different items in the menu.

However, there is no one correct answer regarding the second question. As observed, the menu has two important parts – applicable actions and visualization. This project discusses four actions as part of the simple mark hierarchical marking menu – to draw a mark, to cancel the current mark drawing, to cancel the whole selection, and to perform backing-up. Action backing-up increases the selection speed as well as provides a stable level of error rate. Since actions backing-up and cancel the whole selection are independent from one another and can be presented in the menu at the same time, a design that implements the discussed four actions would be the optimal

menu design. The user would be able to decide which actions correspond to their needs and style of work. The attempt to add sounds at the end of every completed action shows that the sounds slow down the selection process. The reason for this result is the fact that the user waits until hearing the corresponding sound before continuing with the selection of the next level of the menu.

The second important part of the menu – the visualization, can be improved. At this point the menu is static, i.e. it is displayed all the time and at the same location. It might be beneficial to deal with a dynamic menu, so that the user is able to show and hide it at any time they need, as well as to place it on the screen where they feel it would be convenient to use. In addition, the current design deals with novice users only. In future, both novice and expert users should be taken into account.

REFERENCES

1. Kurtenbach, G., Buxton, W. User Learning and Performance with Marking Menus. *Human Factors in Computing Systems*, CHI'94, Boston, USA, 258-264.
2. Zhao, S., Balakrishnan, R. Simple vs. Compound Mark Hierarchical Marking Menus. *UIST'04*, October 2004, Santa Fe, USA, 33-42.
3. Tapia, M., Kurtenbach, G. Some design Refinements and Principles on the Appearance and Behavior of Marking Menus. *UIST'95*, November, 1995, Pittsburgh, USA, 189-195.
4. Zhao, S., Agrawala, M., Hinckley, K. Zone and Polygon Menus: Using Relative Position to Increase the Bredth of Multi-Stroke Marking Menus. *CHI'06*, April, 2006, Montreal, Canada, 1077-1086.
5. Kurtenbach, G., Buxton, W. The Limits Of Expert Performance Using Hierarchic Marking Menus. *INTERCHI'93*, April, 1993, 482-487.