Menu Interactions in a Desktop Haptic Environment

Chris Raymaekers Karin Coninx

Expertise Centre for Digital Media
Limburg University Centre
Wetenschapspark 2, 3590 Diepenbeek
Belgium
Tel. +32 11 26 84 11 fax +32 11 26 84 00
E-mail: {chris.raymaekers, karin.coninx}@luc.ac.be

Abstract

Since the importance of haptic interaction in virtual environments grows, the relevance of good haptic UI widgets becomes more important too. This papers shortly discusses a framework for building 3D widgets in general and more specific 3D menus. The usage of these menus was assessed in a formal user test. The surplus value of using haptic feedback in 3D widgets and the difference in performance of clicking versus pushing were evaluated. This experiment and its results are explained in this paper.

1. Introduction and motivation of the research

Haptic interaction is becoming more important in 3D environments. Since the user typically holds a haptic device, such as the PHANToM [14], in the dominant hand, it is not convenient to use a mouse for menu interaction. Although a minority of the users are capable of manipulating a mouse with the non-dominant hand, it is likely that another 3D input device, such as a space mouse is used for camera navigation. Such a two-handed input metaphor is common in many 3D applications, such as 3D modelling [3, 15]. Furthermore, a menu paradigm should be developed that also supports users who are not bimanual. Therefore, we chose to use the haptic device to interact with a menu. This opens the possibility for placing a 3D menu inside the virtual world.

Although menu interaction has been thoroughly explored in (traditional) VR research [2], the implications and most effective use of 3D menus in a haptic context are not yet known. Different approaches to incorporate menus in a virtual environment can be identified. In JDCAD [6, 8] Spherical and ring menus are used, while pie menus are utilized in HoloSketch [4]. The latter approach can also be found in Alias|Wavefront's Maya [7], where a radial menu, called the Hotbox is placed around the user's mouse cursor. This is, however, a 2D solution. Lindeman et al. [9] used hand-held windows to provide a passive feedback for manipulating 2D widgets in a 3D world with the user's finger.

Active force feedback, using a PHANToM device, has been applied in various 2D desktops. Force feedback has been incorporated in the X desktop by Miller et al. [10, 11] while Oakley et al. enhanced the Microsoft Windows desktop [12]. Some research into 3D haptic user interfaces has has also been conducted: Anderson et al. [1] have created a interface builder for graphical and haptic user interfaces (GHUIs) for FLIGHT (FLIGHT is now called e-Touch [5]) in which several interfacing components are present. Since, to our knowledge, no research exists in which the usefulness of haptic menus in investigated, we developed a 3D haptic menu and performed a formal user experiment in order to assess the effectiveness of our solution.

This paper presents a technical overview of the menus that we provide in our haptic environments and discusses the user experiment.

2. Haptic interfacing elements

All 3D interfacing elements, such as menus, toolbars and dialogs, have to provide a common functionality. For instance, they have to support the haptic device and should allow the developer to place descriptive texts on them. We call these objects “haptic UI elements”.

2.1. Placement

Another problem with 3D UI elements is the fact that they sometimes obscure the element of interest: a dialog, which asks to confirm the deletion of an object, should not be placed in front of this object. Likewise, if an object

Figure 1. Semitransparent menu

Figure 2. Opaque menu

were to obscure a haptic UI element, the haptic UI element would be unusable. Therefore, a haptic UI element should be placed rather in the foreground of the virtual environment than in the background. However, it must not hinder the user in interacting with the virtual objects. Hence, we have developed haptic UI elements that can arbitrarily be positioned in the virtual environment, but are typically placed in the foreground. The haptic UI elements are semitransparent, thus not obscuring the virtual objects [16]. When the virtual pointer approaches the haptic UI element, it fades in to become opaque. As the pointer moves away, the haptic UI element fades out again. Figures 1 and 2 show a menu in its semitransparent and opaque form.

2.2. Implementation

In order to use haptic UI elements in a virtual environment, we have written an abstract C++ class, which supports the above-mentioned functionalities for 3D UI interaction. When the first haptic UI element is instantiated, it makes a series of OpenGL display lists in which the alphabet is stored. All haptic UI elements can then use these display lists to display their textual contents. Likewise, the abstract class is responsible for placing a haptic constraint in the virtual environment. This haptic constraint represents the haptic UI element and calculates where the virtual pointer, representing the haptic device's position and orientation, is located with respect to the haptic UI element.

Currently, a class is derived from the abstract class, which implements a 3D menu: a 2D menu object, which can be arbitrary positioned in 3D space. Each menu item in this 3D menu is indicated by a text, e.g. "Exit". Optionally, a descriptive icon can be depicted in front of the menu item. In order to make the menu recognizable for the user, it employs the Windows colour scheme. Two different interaction methods are provided by the menus: "point and click" and pushing against the menu item. The first method makes use of the standard method for accessing 3D menu's, although a 3D pointer is used in this case, while the latter is based on real life interaction with buttons and switches.

3. User experiment

The usefulness of the menu interactions was assessed in a formal user experiment. We wanted to assess if the "point and click" metaphor is better than the push metaphor and if force feedback improves menu selection. A scene representing a room, containing a cube and menu, was shown to the users. The menu items indicated 7 colours; the name of each colour was preceded by a square in the same colour, so that the user could easily match the colour of the cube with a menu item. We choose to add this square instead of depicting the text in the colour it describes, because not all colours are equally readable. Figure 3 depicts the experimental scene; since our test persons were native Dutch speakers, the colours are indicated in Dutch. The test subjects had to indicate the current colour of the cube, which was randomly chosen, in the menu. The performance of the users when using haptic feedback, provided by a PHANToM device, was compared with a condition where no haptic feedback was present. In each condition, a test person was presented with 5 practice trials and 15 measured trials.

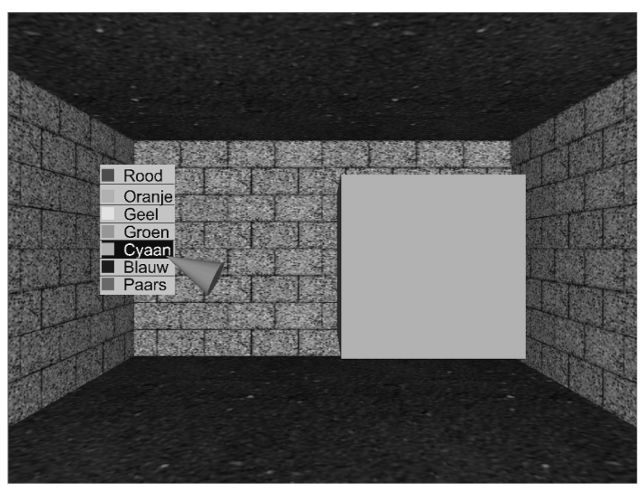

Figure 3. Experimental scene

Twenty-four test persons, twenty males and four females with an average age of 30, participated in a counterbalanced repeated measures design. In order to avoid negative transfer effects, the test persons where evenly distributed into two groups according to age, sex and experience with computers. One group of 12 test subjects was presented with the "point and click" interaction (click condition), while the other group had to push against the menu items (push condition).

The dependent variables we measured were: elapsed time (in ms), needed to select a menu item, the distance covered by the virtual pointer (in mm) and the number of erroneous selections. The elapsed time was further divided in the approach time, needed to come in the vicinity of the menu and the homing time, needed to select the item within the menu. A trial was considered complete, when the correct menu item was selected. After completing the test in each particular condition the subjects were asked to answer a series of questions. After completing the second condition, a comparing questionnaire was presented. This questionnaire presented the test persons some questions about their subjective feeling of performance and frustration.

4. Results

The results for both conditions were analysed using two-way ANOVA over all measurements (click and push conditions). This analysis reveals that the use of force feedback only significantly reduces the number of errors that were made by the test subjects. When no force was applied, the results were better for all other dependent variables, though not significantly. Table 1 summarizes these results.

Table 1. Statistical analysis of all results

	Without force	With force	P-value
Avg. approach time	2053.7	2112.1	.86
Avg. homing time	3167.4	4076.5	.14
Avg. total time	5221.1	6188.6	.11
Avg. distance	60.7	74.8	.11
Total errors	150	68	$< \mathbf{.001}$

Furthermore, the trade-off between precision and speed is confirmed by our results, since a negative correlation between the approach and homing time can be found (see table 2).

However if the two selection mechanisms are compared, a significant difference can be found in favour of the clicking with the stylus switch (see table 3). At first sight, this

Table 2. Correlations

	Avg. approach time	Avg. homing time	Avg. distance
Avg. approach time	1.0		
Avg. homing time	-0.27	1.0	
Avg. distance	0.27	0.76	1.0
Total errors	0.06	0.40	0.35

seems to be contra intuitive: in real-life, selecting a button, e.g. a button on a microwave oven, is done by pushing against the button, not by clicking with another device. However, in real-life people push against buttons with their fingers, not with a pen, while people are used to the "point and click" interaction when working with a computer. As an alternative, we could have chosen to use the PHANToM thimbal, because it allows the user to push against objects with the index finger. However, most of our applications require 6DOF input, which can only be provided with the encoder stylus. Furthermore, most 3D interaction benefits from having a switch on the encoder stylus (e.g. to select objects in the virtual world).

Table 3. Comparing the click with the push condition

	Click	Push	P-value
Avg. approach time	1387.6	2778.2	$< \mathbf{.001}$
Avg. homing time	1894.9	5348.9	$< \mathbf{.001}$
Avg. total time	3282.5	8127.1	$< \mathbf{.001}$
Avg. distance	42.7	92.8	$< \mathbf{.001}$
Total errors	79	137	$< \mathbf{.001}$

Within the click condition, no significant difference was found in the time needed to complete the condition with haptic feedback and the condition without haptic feedback. The approach time was slightly better when no haptic feedback was present. However, this was compensated by the better homing time in the condition with haptic feedback, leading to better over all performance in the condition with haptic feedback. Video recordings, from the user test, suggest that this difference is caused by a slight hesitation when approaching the haptic menu because the test subjects searched the haptic plane, whereas in the condition without haptic feedback, the test subjects just located the pointer in the vicinity of the menu. However, as soon as the virtual pointer hit the haptic plane, the test subjects could use this plane as a guide and worked faster and more precise. The fact that haptic feedback aids the user in being more efficient is supported by the results of the error measurements (see table 4). In both cases 180 (12 test persons x 15 trials) correct clicks were recorded. In the condition without haptic feedback another 63 erroneous errors were made; hence, 26% of all clicks were wrong. Haptic feedback, however reduces this figure: 8.2% of all clicks (16 out of 196) were erroneous. This result is statistically significant. We reckon that this difference is caused by test persons, who overshoot the virtual pointer when no haptic feedback is present. Furthermore, if the encoder stylus does not rest against a virtual plane, clicking on the switch will sometimes cause the stylus to move downwards before the click is registered.

Table 4. Analysis of the click condition

	Without force	With force	P-value
Avg. approach time	1365.3	1409.8	0.82
Avg. homing time	2038.4	1751.4	0.35
Avg. total time	3403.8	3161.3	0.41
Avg. distance	42.8	42,5	0.94
Total errors	63	16	$< \mathbf{.001}$

No significant difference was found in the results of the post experiment questionnaire, although a number of test subjects in the push condition were visibly frustrated during the experiment.

5. Discussion

Our experiment shows that selection of a menu item in a haptic scene can be best performed by using a “point and click” metaphor. We believe that these results are valid for a number of selection widgets, such as toolbars and lists. In our setup, pointing can be accomplished in a reliable and fast manner. It is reliable because the menu item that can be selected lights up when the pointer is near to the item (but this is also true for the pushing mechanism). It is also fast, because the user does not have to touch the menu, lighting a menu item by coming in the vicinity is good enough. However, the menu lets the users rest their hand against a haptic plane, which helps to reduce errors. We noticed both behaviours during the test.

Of course, the haptic case still does not provide perfect selection, so usual error correction mechanisms, such as a confirmation for critical operations (e.g. clearing the virtual scene) and an undo mode for other operations, must be provided.

A problem that arose, when working with the haptic menu was the fact that it was only touchable from the front. We allowed to push trough the menu, when approaching it form the back in order to enhance the push condition. Sometimes, the test subject would get lost and ended up behind the menu. In order to push against the menu, the user must be in front of it. In this case the test person would have to go round the menu with the pointer If the back of the plane would also be touchable. We choose to keep this effect in the “point and click” condition, in order to minimize the differences between both conditions. The test persons in the push condition where happy with this feature, but the test persons in the “point and click” condition that ended up behind the menu reported that they would be more efficient if the menu would be touchable from two sides. This is an extra argument in favour of the “point and click” condition, since the use of a menu that is touchable from both sides is consistent with a haptic virtual environment. However, further research is needed to support this thesis.

Although the usage of semi-transparency allows us to place the menu in an arbitrary location, it would be better if it would not be visible if the user does not want to use the menu. We have already explored the use of head-tracking in a desktop virtual environment [13]. With this setup, users can virtually enlarge their workspace and can look at the virtual world from different viewpoints in an intuitive manner. If a menu would be placed just outside of the users view, than the menu can be easily accessed without being intrusive. Of course, semi-transparency would still be desirable, because it is still possible that the user just wants to examine an object.

6. Conclusions

As a first conclusion, we can state that haptic 3D menus in a virtual environment should use the “point and click” metaphor for selection and not a pushing metaphor when using an encoder stylus, although the latter seems to be obvious from real-world experiences. Secondly, haptic feedback is invaluable in this interaction, in order to reduce the number of errors that arise due to overshooting. We believe that similar conclusions can be drawn for other haptic UI elements, such as dialogs and toolbars, although further research is needed to support this hypothesis.

7. Acknowledgements

Part of the work presented in this paper has been funded by the Flemish Government and EFRO (European Fund for Regional Development). We would also like to thank Wouter Kempen (graduate student at the University of Maastricht who made his thesis at the EDM-LUC) for his help in conducting the user experiments.

References

[1] T. Anderson, A. Breckenridge, and G. Davidson. FBG: A graphical and haptic user interface for creating graphical, haptic user interfaces. In *Proceedings of the Fourth PHANToM Users Group Workshop, October 9–12 1999, Dedham, MA, USA*, 1999.

[2] K. Coninx, F. Van Reeth, and E. Flerackers. A hybrid 2d/3d user interface for immersive object modeling. In *Proceedings of Computer Graphics International '97, Hasselt and Diepenbeek, Belgium, June 23–27 1997, Denver, CO, USA*, pages 47–55, 1997.

[3] J. De Boeck, C. Raymaekers, and F. Van Reeth. Introducing touch in a rigid body simulation environment: a design overview. In *Proceedings of PURS 2000, Selected Readings in Vision and Graphics, Vol. 8, 6–7 july 2000, Zurich, CH*, pages 1–8, 2000.

[4] M. Deering. The holosketch VR sketching system. *Communications of the ACM*, 39(5):54–61, May 1996.

[5] e-touch. www.etouch3d.org.

[6] M. Green and S. Halliday. A geometric modeling and animation system for virtual reality. *Communications of the ACM*, 39(5):46–53, May 1996.

[7] G. Kurtenbach, G. Fitzmaurice, R. Owens, and T. Baudel. The Hotbox: Efficient access to a large number of menu-items. In *Proceedings of CHI 2000, April 1–6, Den Haag, NL*, pages 231–237, 2000.

[8] J. Liang and M. Green. JDCAD: A highly interactive 3d modeling system. *Computers & Graphics*, 18(4):499–506, 1994.

[9] R. W. Lindeman, J. L. Sibert, and J. K. Kahn. Hand-held windows: Towards effective 2d interaction in immersive virtual environments. In *Proceedings of IEEE Virtual Reality '99, March 13–17 1999, Houston, TE, USA*, pages 205–212, 1999.

[10] T. Miller. Implementation issues in adding force feedback to the X desktop. In *Proceedings of the Third PHANToM Users Group Workshop, October 3–6 1998, Dedham, MA, USA*, 1998.

[11] T. Miller and R. Zelevnik. An insidious haptic invasion: Adding force to the X desktop. In *Proceedings of the 11th ACM UIST'98 symposium on user interface software and technology, ACM, San Francisco, Ca, USA, Nov 1–4 1998*, pages 59–66, 1998.

[12] I. Oakley, M. R. McGee, S. Brewster, and P. Gray. Putting the feel in 'look and feel'. In *Proceedings of CHI 2000, April 1–6, Den Haag, NL*, pages 415–421, 2000.

[13] C. Raymaekers, J. De Boeck, and K. Coninx. Assessing head-tracking in a desktop haptic environment. accepted for HCI International, August 5-10 2001, New Orleans, LA, USA, 2001.

[14] SensAble. www.sensable.com.

[15] R. Stoakley, M. Conway, and R. Pausch. Virtual reality on a WIM: Interactive worlds in miniature. In *Proceedings of CHI '95, May 7–11, Denver, CO, USA*, pages 265–272, 1995.

[16] S. Zhai, W. Buxton, and P. Milgram. The partial occlusion effect: Utilizing semitransparency in 3D human-computer interaction. *ACM Transactions on Human-Computer Interaction*, 3(3):254–284, 1996.