Examensarbete

Advanced Multi-modal User Interfaces in 3D Computer Graphics and Virtual Reality

Examensarbete utfört i Advanced Computer Graphics vid Tekniska högskolan i Linköping
av

Yenan Chen

LiTH-ITN-TEK-A--12/014--SE
Norrköping 2012
Advanced Multi-modal User Interfaces in 3D Computer Graphics and Virtual Reality

Examensarbete utfört i Advanced Computer Graphics vid Tekniska högskolan i Linköping
av

Yenan Chen

LiTH-ITN-TEK-A--12/014--SE

Handledare: Matthew Cooper
ITN, Linköpings universitet

Examinator: Karljohan Lundin Palmerius
ITN, Linköpings universitet

Norrköping, 6 March, 2012
Abstract

Computers are developed continuously to satisfy the human demands, and typical tools used everywhere for ranging from daily life usage to all kinds of research. Virtual Reality (VR), a virtual environment simulated to present physical presence in the real word and imaginary worlds, has been widely applied to simulate the virtual environment. People’s feeling is limited to visual perception when only computers are applied for simulations, since computers are limited to display visualization of data, while human senses include sight, smell, hearing, taste, touch and so on. Other devices can be applied, such as haptics, a device for sense of touch, to enhance the human perception in virtual environment. A good way to apply VR applications is to place them in a virtual display system, a system with multiply tools displays a virtual environment with experiencing different human senses, to enhance the people’s feeling of being immersed in a virtual environment. Such virtual display systems include VR dome, recursive acronym CAVE, VR workbench, VR workstation and so on.

Menus with lots of advantages in manipulating applications are common in conventional systems, operating systems or other systems in computers. Normally a system will not be usable without them. Although VR applications are more natural and intuitive, they are much less or not usable without menus. But very few studies have focused on user interfaces in VR. This situation motivates us working further in this area. We want to create two models on different purposes. One is inspired from menus in conventional system and the sense of touch. And the other one is designed based on the spatial presence of VR.

The first model is a two-dimensional pie menu in pop-up style with spring force feedback. This model is in a pie shape with eight options on the root menu. And there is a pop-up style hierarchical menu belongs to each option on the root menu. When the haptics device is near an option on the root menu, the spring force will force the haptics device towards to the center of the option and that option will be selected, and then the sub menu with nine options will pop up. The pie shape together with the spring force effect is expected to both increase the speed of selection and decrease the error rate of selection. The other model is a semi-automatic three-dimensional cube menu. This cube menu is designed with a aim to provide a simple, elegant, efficient and accurate user interface approach. This model is designed with four faces, including the front, back, left and right faces of the cube. Each face represents a category and has nine widgets. Users can make selections in different categories. An efficient way to change between categories is to rotate the cube automatically. Thus, a navigable rotation animation system is built and is manipulating the cube rotate horizontally for ninety degrees each time, so one of the faces will always face users. These two models are built under H3DAPI, an open source haptics software development platform with UI toolkit,
a user interface toolkit.

After the implementation, we made a pilot study, which is a formative study, to evaluate the feasibility of both menus. This pilot study includes a list of tasks for each menu, a questionnaire regards to the menu performance for each subject and a discussion with each subject. Six students participated as test subjects. In the pie menu, most of the subjects feel the spring force guides them to the target option and they can control the haptics device comfortably under such force. In the cube menu, the navigation rotation system works well and the cube rotates accurately and efficiently. The results of the pilot study show the models work as we initially expected. The recorded task completion time for each menu shows that with the same amount of tasks and similar difficulties, subjects spent more time on the cube menu than on the pie menu. This may implicate that pie menu is a faster approach comparing to the cube menu. We further consider that both the pie shape and force feedback may help reducing the selection time. The result for the option selection error rate test on the cube menu may implicates that option selection without any force feedback may also achieve a considerable good effect. Through the answers from the questionnaire for each subject, both menus are comfortable to use and in good control.
Acknowledgments

I would like to thank my examiner Karljohan Lundin Palmerius, my supervisor Matt Cooper and my parents.
Contents

1 Introduction
   1.1 Virtual Reality And VR Workbench .......................... 2
   1.2 User Interface in Virtual Reality ............................ 4
   1.3 H3DAPI ......................................................... 5
      1.3.1 Level programming ..................................... 6
      1.3.2 Toolkits .................................................. 6
   1.4 Overview of the Project ....................................... 7
      1.4.1 Purpose ................................................... 7
      1.4.2 Goal ........................................................ 7
      1.4.3 Limitations ............................................... 8

2 Theory Preliminaries .................................................. 9
   2.1 2D Pie Menu ................................................... 9
      2.1.1 'Pie' shape .............................................. 10
      2.1.2 Force feedback .......................................... 10
      2.1.3 Popup submenu .......................................... 11
   2.2 3D Cube Menu .................................................. 12
      2.2.1 Rotation .................................................. 12
      2.2.2 Accuracy test .......................................... 14

3 Implementation ....................................................... 15
   3.1 The Implementation Overview .................................. 15
   3.2 2D Pie Menu ................................................... 16
      3.2.1 Structure ................................................ 16
      3.2.2 Menu category .......................................... 17
      3.2.3 Spring effect .......................................... 18
   3.3 3D Cube Menu .................................................. 19
      3.3.1 Structure ................................................ 19
      3.3.2 Menu category .......................................... 19
      3.3.3 Rotation ................................................ 20
      3.3.4 Accuracy test .......................................... 23
Chapter 1

Introduction

Computers are the typical tools for visualizing data. The range of the visualization is widely from statistical charts to medical visualization. As the demand for research is continuously higher, the research is not satisfied only with human vision, but also with some other human senses, such as sense of touch and sound. Virtual Reality (VR) provides such a possibility to increase the awareness of features in data[14]. Virtual Reality is more and more used in research and developing applications, especially haptics and multimodel applications in various fields including but not limited to dental, medical, industrial and visualization.

Though Virtual Reality is very different from conventional computer systems, and can make tasks more natural and simpler, still it is necessary to add conventional elements such as menus to make applications more usable and efficient. This motivates us working further in the user interface area in Virtual Reality, to research and implement menus that are inspired and developed from previous work for future research and real applications. The diploma topic is proposed based the background described above. This diploma work is a research in the area of user interfaces in Virtual Reality. A multi-model menu system is designed and implemented, a report is written to elaborate what has been done during this period. We want to create two models on different purposes. One is inspired from user interfaces in conventional system and we combine the speciality of Virtual Reality to the model. The other one is based on the spatial presence of Virtual Reality. These two models are then implemented. One is a two-dimensional (2D) pie menu in pop-up style with spring force feedback on option selection, and the other one is a semi-automatic three-dimensional (3D) cube menu. Although there are limitations on these two styles of models, for example, with equal amount of options, pie menu takes more space than ordinary menus. And 3D cube menu has limitation on manipulating the menus movement, rotation is problem since users could not see all the options at the same time. Some limitation will be fixed in the work, while some could not. Since these two models are rather different and relatively complemented, we could choose the other model if one is facing certain limitations. The goal of project is to explore menus with simple but elegant interface to achieve effective operation and gain fine control of the machine. The goal
of the 2D pie menu is to achieve good force feedback to improve the speed and decrease the error rate. The goal of the 3D cube menu is to navigate the menu movement correctly and rotate smoothly. In conclusion, the result of the project reached the goal. The pie menu is implemented with a fast and low error rate approach, while the cube menu is spatial and navigable and derives good visual effect.

This report will cover what the diploma work has been done. This chapter will elaborate the background of the thesis, and software and hardware environment for the menu system, related research on user interface in virtual environment, the purpose, goal and limitation of the work. The following chapter will describe the theory preliminaries of the 2D pie menu as well as 3D cube menu in the study. In the third chapter, the implementation process of these two menus will be stated respectively and described in detail. In chapter 4, a pilot study evaluating the implementation is made. Final conclusion and suggestion for the future improvement are concluded in chapter 5.

1.1 Virtual Reality And VR Workbench

Virtual Reality (VR) is one of the key concepts in the project. It applies to computer-simulated environments to simulate physical presence in the real world and in imaginary worlds. Most current virtual environments are mainly in visual experiences, displayed either on a computer screen or through special stereoscopic displays. Typically, the visual display is a 3D representation of a world rendered using 3D graphics. More senses of human perception are added to the virtual environment, as the techniques used for acquiring data are becoming increasingly advanced. in Virtual Reality, the multi-senses of human perception that applied give users a sensation of being immersed in an environment. The sensory simulation could sound through speakers or headphones. Even more advanced, haptic simulation, it includes tactile information such as force feedback. Such simulation is typically used in medical and gaming applications [24, 15].

The Virtual Reality is not real, but can be similar to real world in order to provide a lifelike experience, and is better than real things because it is not hazardous when doing dangerous things and the work is not constrained by telepresence, scaling and etc. The Virtual Reality system is better than other systems in the aspects that it is intuitive to use and is full body control with speed and accuracy and it is possible to use multiple sensors. It also has advantages in spatial understanding, natural and intuitive navigation, increased comprehensibility and engagement compared to other work in different environments [15].

In a paper by Krueger [10], they presented the interpretation of Virtual Reality workbench (VR workbench), it is a virtual working environment that locates virtual objects and control tools on a real workbench. The stereo images generated from computer are projected onto the screen of a workbench and displayed properly through the screen and stereo shutter glasses. The workbench can integrate various input and output modules, such as haptics, motion, gesture, and voice recognition systems. It is possible for several people to work together in sim-
ilar environments either locally or by using broadband communication networks. Such a responsive environment, consisting of powerful graphics workstations, with big back-projected large screen, stereo, head tracking device and wand/stylus, LC shutter glasses, is a typical VR workbench in Virtual Reality. Our project works in such an environment. Four devices in the VR workbench are applied to use. They are back-projected large screen, IS-900 system for head tracking, LC shutter glasses and a PHANToM stylus.

**Liquid crystal shutter glasses (LC shutter glasses):**

![Shutter glasses](image)

*Figure 1.1. Shutter glasses*

Introduced by [20], the shutter glasses together with a display screen creates the illusion of a 3D image. Each eye’s glass contains a liquid crystal layer which contain property of becoming dark when voltage is applied, being otherwise transparent. The glasses are controlled by an infrared or other transmitter that sends a timing signal to allow glasses to alternately darken over one eye and the other, synchronized with refresh rate of the screen. At the same time, the display alternatively displays different perspectives for each eye.

**Intersense IS-900 system:**

![Headtracking device](image)

*Figure 1.2. The Headtracking device of the Intersense IS-900 System*

Cited from InterSense Incorporated [6], the IS-900 is the system for precise position and orientation, referring to six degrees of freedom (6DOF) tracking in
military and industrial simulators, immersive displays, virtual prototyping and film/video production. 6DOF refers to the ability to move forward/backward, up/down, left/right namely translation in three perpendicular axes combined with pitch, yaw and roll about three perpendicular axes [21]. Based on InterSense’s patented inertial-ultrasonic hybrid tracking technology, the system is immune to metallic interference, while offering smooth, robust real-time tracking in demanding environments. The system used in the study is configured with a MicroTrax Head Tracker, tracked with fast update rates, low latency with no jitter, for use with stereo glasses for interaction with the virtual content. The inertial tracking technology senses user’s head movements, and the sensor fusion software which includes motion prediction and perceptual enhancement, is able to predict where the user intend to look and adjust the virtual display accordingly.

**PHANToM Premium 3.0:**

![PHANToM Premium 3.0](image)

This haptics device provides a range of motion approximating full arm movement pivoting at the shoulder. It provides three degrees of freedom positional sensing and three degrees of freedom force feedback [17].

### 1.2 User Interface in Virtual Reality

The user interface (UI), is designed for interaction between humans and machines occurs in the field of human-machine interaction. User interface used in interaction
1.3 H3DAPI

is aim to provide effective operation and fine control of the machine [23]. Conventional computer system cannot be usable without the user interface. And the user interface has been developed rapidly since there has been a tremendous growth in interest about human computer interaction in the past years. The Menu, a sequential list of text, icons or other graphics items arranged in a geometry format, as the main part of user interface, follows the same trend. Although more and more research and applications are applied in Virtual Reality, still not many researches are concerned on the user interface area especially on menu design.

In Virtual Reality, Good and complicated applications could not be effectively manipulated without a well designed user interface which can make human-machine interaction easy, efficient and enjoyable to operate and produce desired results. Our study is aim to research, implement and analysis the characteristics of two user interfaces for practical application, future study and other researches in related areas.

We put formats, display methods, shape and dimension into consideration when we design the menu. Rectangle is a typical format, there are also other formats used often, such as semi-circle, circle and so on. The display methods are sorted from pop-up, pull-down, stack menu, object-specific menu to oblique/layered menus; it ranges from usual to compact ones to address hierarchical menu system in Virtual environment (VE) [8]. Sorted by shape, there are linear menu, ring menu, pie menu, finger menu and so on. The menu is sorted to 2D and 3D menu by dimension. Different menu structures are applied based on different situations. If the application is simple, one single menu with several options may suffice. For example, ring menu introduced by [5], that acts as a contextual or global menu arranged on a part of a circle and controlled by rotating the wrist in the horizontal plane. If there are more related options than will fit on a single menu, a hierarchical menu is useful [12]. Such a hierarchical menu usually shows up with a menu bar with several options, where each option has a group of commands associated with a particular aspect of the application [18].

To make option selection more efficient, some methods can be applied to the menu, such as changing the color or size of the button, and adding haptics feedback to the button. Adding force feedback is a special method in Virtual Reality comparing with in conventional systems, since no sense of touch can applied to conventional systems. Various haptics feedback on buttons have been applied, such as menus with 2D planar constraint, menu with edge boundary constraints feedback and so on [9].

1.3 H3DAPI

In order to implement the Virtual Reality application, we have to choose a well performed haptics software development platform. CAD, JAVA 3D, H3DAPI, and other software platforms provide the different ways to write a wide variety of Virtual Reality applications. The application for this diploma work is implemented under H3DAPI, an open source haptics software development platform that is written in C++ and designed to be extensible. Here list the reasons for us to
choose H3D API as the platform to implement the project. H3DAPI uses the open standards OpenGL and X3D with haptics in one unified scene graph to handle both graphics and haptics. It is a cross-platform API and designed mainly to support a special rapid development process. It combines X3D, C++ and the scripting language Python to offer three ways of programming applications and is the ideal tool for writing hapto-visual applications that combine the sense of touch with vision. There are many toolkits developed with it, especially UI toolkit with robust ability to creating menu framework. H3D is easily customized to work with a wide variety of VR display systems primarily aimed at the SenseGraphics line of immersive workstations. It supports PHANToM devices which we used in the diploma work, Force Dimension devices and Novint Falcon.

1.3.1 Level programming

As it is mentioned before, there are three levels for H3D Programming - X3D, Python or C++. X3D and Python provide high level interfaces to API, while C++ provides raw access to API. More description presented by H3DAPI manual [1] are as follows.

X3D is a scene-graph concept data structure program, it is divided to Nodes and Fields. Fields are data containers that store and manipulate data properties. Nodes contain and manage fields, and are implemented though fields. X3D file format defines geometries and arranges scene-graph elements in a most easy way, are the reasons that X3D used by H3D. Compared to C++, X3D is easy to read, maintain and debug. H3D has a full XML enables X3D parser and supports predefined X3D nodes and new nodes developed for the application.

Python is an interpreted language so that the Python script does not have to be compiled to run. Fields that controls simple non-time-critical behavioral properties such as managing a user-interface could be ideally defined by Python. Also, Python can be effective for prototyping complex time-critical fields that can be ported to C++ after prototyping.

C++ is a statically typed, free-form, multi-paradigm, compiled, general-purpose programming language. It is regarded as an intermediate-level language, as it comprises a combination of both high-level and low-level language features [16]. It can be used to create highly efficient code. It is ideal to write haptic rendering algorithms or use OpenGL with C++. But it has compile-test-debug cycle which means finding and fixing bugs can often take long time. So, if it is time critical, the function should be encapsulated as a C++ node or field in the API.

1.3.2 Toolkits

Many toolkits are developed with H3D for different uses, such as Volume Haptics Toolkit for bringing haptics into volume data exploration interfaces and the volume data understanding process, Candy for providing a collection of miscellaneous tools for example the calibration tool, MedX3D for a Volume Rendering component of X3D, UI toolkit for haptic 3D-widgets and so on. The menu framework is typically built by applying the UI toolkit. It contains nodes for creating widgets, such as
1.4 Overview of the Project

This section gives more detailed descriptions on purpose, goal and limitations of the project.

1.4.1 Purpose

So far, only a limited number of menu styles have been provided in Virtual Reality. Most of them are 2D menus and have been applied a lot in conventional systems, such as linear menu and rectangular menu. And 3D menus, which puts more spatial information to practical use than does a 2D menu, have been researched very few for the usability. Therefore we want to delve into both 2D and 3D menus to provide and advance the user interfaces in Virtual Reality.

1.4.2 Goal

This part describes the goal for the thesis. As mentioned before, we want to design a 2D menu that is developed from one of those menus have been applied in conventional system. Pie menu is a good option. A typical example of the pie menu is the one implemented in Mozilla based Firefox web browser, with eight menu options on the circle and with sub menus that are grouped together from related menu functions. We aim to implement a pie menu with a fast and reliable selection style. We want to make is adding spring force feedback to each option on the circle and achieve good control of the spring force feedback for a list of references. Firstly, users must feel comfortable to control the menu with the added force. Secondly, the spring force feedback must guide the user to select the target correctly. Thirdly, by applying such force feedback, option selection time must be reduced.

Virtual Reality applications are mostly applied in 3D space, this motivates us design a 3D menu. While the 3D cube menu we are interested has been implemented very few so far. We aim to find a good solution for rotating animation control on the cube menu as it is described below. Firstly, with the manipulation on a PHANToM stylus, the cube should rotate smoothly towards either to the left or to the right based on the users’ movement. Secondly, it should rotate ninety degrees each time and could rotate continuously as many times as the user wanted. Thirdly, the rotation should happen around $y$ axis in horizontal plane. Lastly, the desired side of the cube is always facing the user.
1.4.3 Limitations

This part follows the limitations concerned on both the 2D pie menu and the 3D cube menu.

In the pie menu, limited number of options can be placed on it. If there is a need to place more options, those options might need to be placed on more circles on the root menu which consumes much more space than ordinary menus. This would influence the usability of the user interface. And adding the spring force feedback could increase the complexity of user interface. Another concern is a user interface may be complex for one user but for other less or not at all. And It may be much complex at start but almost simple after its prolonged use.

In the cube menu, the users have limited view over the cube. Only one side of the cube is facing the users while the cube is composed by four sides, users might really get confused with the left unseen sides.
Chapter 2

Theory Preliminaries

Menu can be found in everywhere, such as Automatic Telling Machine, restaurant. It is not an exception in Virtual Reality. Adding menu to Virtual Reality application is potentially making operation effective and control of the machine with good effect. Experiences from using conventional systems and learning from papers and even inspirations from science fiction movies give us an overall idea of the implementation in this thesis. The two models we implemented are a 2D pie menu with spring force feedback on buttons and a 3D cube menu with navigable rotation animation.

In the following sections, principles, theory basis and challenges for designing the pie menu and cube menu will be discussed in detail. There are some aspects in common for both of the menu. They are oriented to present a face-on view to the user in the 3D environment. Buttons on menus are sized to contain legible text. There is haptical enhancement on buttons on both menus, e.g. 2D planar force on all buttons. The key challenge for 2D pie menu is how to add force feedback on button properly, and for 3D cube menu is how to correctly navigate the menu rotating smoothly. We will then describe and analyzed them and different parts concerned on each type of menu, such as "pie" shape, popup submenu style, accuracy test in detail.

2.1 2D Pie Menu

The pie menu, a circular context menu where selection depends on direction, first came out in 1969. It now can be found in many contexts, such as a plug-in for firefox for the control of surfing. Researches have shown this menu style to be faster and less error prone than linear menus due the large selection regions and short, consistent movement needed to make a selection [2, 9]. Proper additional techniques on pie menu can even improve the performance. However, a key disadvantage with pie menus is that only a limited number of menu options, typically eight or less can be comfortably shown on a single menu. Since other features of the pie menu are considered weigh over than the number issue, pie menu is still the first option for implementation.
This part elaborates the theory of the pie menu implementation including pie shape, spring force feedback on buttons of main menu, and hierarchical submenus belonging to each button on the main menu.

2.1.1 "Pie" shape

A pie menu is a format where items are placed equally along the circumference of a circle. It gains over traditional linear menus by reducing target seeking time, lowering error rates by fixing the distance factor and increasing the target size, and is, in general, subjectively equivalent to the linear style. Previous researches have been showing the benefit of pie menus over list of 2D menus in option selection that pie menu is faster, reliable, easy and efficient, e.g., [3, 9]. The pie menu with selection methods and assistive forces is stated to perform considerably better than linear menus with the same additional elements, in the aspects of efficiency and accuracy [9].

![Pie shape](image)

Figure 2.1. Pie shape

Basically, pie menus utilize a circular design where options are arrayed in segments on the circle and provide a smooth and reliable gestural style of interaction [5], see figure 2.1. In our implementation, eight standard cubic square buttons are around the circle, to make a selection a user moves PHANToM stylus to one of the buttons, and then selects by pressing it.

2.1.2 Force feedback

Recent studies have showed that force feedback can provide improvement in the menu selection process for pie menu. For example, Rick and Colin [9] carried out a study where subjects made selection from two styles of linear menu and a pie menu with eight options for each, using PHANToM as the haptic device. The option selections techniques applied are 'release button', "push-through" and "exceed border option interaction". And three assistive force components in menu option are studied. It was expressed that, option selection techniques on pie menu
2.1 2D Pie Menu

performed much better than on straight and slant linear menu. And it is shown [13], with spring force, task completion time and error rates are comparable to or better than the purely visual case.

Thus, apart from the planar force feedback on each buttons, we decide to use the technique of spring force feedback in option selection on pie menu implementation. The spring force applied in the thesis follows the Hooke’s law. In mechanics, and physics, Hooke’s law of elasticity is an approximation that states the extension of a spring is in direct proportion with the load applied to it [19]. The strain is directly proportional to stress in Hooke’s law.

![Hooke’s law graph](image)

Figure 2.2. Hooke’s law

Mathematically, Hooke’s law states that

\[ F = -kx \]

where

- \( x \) is the displacement of the spring’s end from its equilibrium position (a distance, in SI units: meters);
- \( F \) is the restoring force exerted by the spring on that end; and
- \( k \) is a constant called the rate or spring constant.

Hooke’s law is valid throughout its elastic range. Many materials obey this law as the load does not exceed the elastic limit. When the spring holds, the behavior is linear. The restoring force always acts in the opposite direction of the displacement since there is a negative sign on the right hand side of the equation. So the action of a spring is when a spring is stretched to the left, it pulls back to the right [19].

In the pie menu, the spring force on each button follows the rules elaborated above, the elastic range is distance from spatial border to the center of the button, and the spatial border could be spherical, cubic or other figurate. When the PHANToM is on the border, the force reaches its biggest. And when the PHANToM is at the center of the button, the force turns to 0.

2.1.3 Popup submenu

Usually, there are more than several options in the menu, as there is a limited set of options on the main menu, and more available choices are actions related to the
each option on the main menu. In such case, the set of available choices related to one option can be in a submenu. In the implementation, the submenus of the pie menu are hierarchical to the selected objects on the main menu. Pop-up is the most popular and traditional style that used in hierarchical menu design, and is utilized in our application. The hierarchical submenu with nine related options formatted in square pops up on the right side of the root menu when it is selected. If that button is pressed again or any other buttons on pie menu is pressed, the submenu disappears or another hierarchical submenu pops up.

2.2 3D Cube Menu

In nowadays, option selection interaction in systems largely depends upon 2D user interfaces. In the following part, we explore an alternate paradigm - 3D cube menu as another fundamental interaction method. With its spatial presence, 3D user interfaces support user tasks in many non-conventional systems such as virtual environments and augmented reality. They have been designed and applied to improve the usability of the applications in Virtual Reality for example a virtual widget. Usually, 3D menus are more of attracting users’ interest than 2D menus by applying intuitive menu navigating interaction. The 3D cube menu presented by us is a good example with its novel interface and the accurate navigating rotating system. And this menu is with simple, elegant, space saving, spatial representative and intuitive manipulation. Though, only limited view of the cube menu is presented to the user each time and limited options, such drawbacks will not influence the performance in the report and can be improved in the future.

This cube menu is designed with four faces. Those four faces are the front, back, left and right face of the cube. Each face represents a category and has nine widgets. A navigable rotation animation system is manipulating the cube rotate horizontally each time for ninety degrees. The goal of the rotation system is automation, intuitivity, smoothness and accuracy. An accuracy test is presented at last for usability evaluation.

2.2.1 Rotation

If the 3D cube rotates exactly as the PHANToM's movement, apparently its action is not automatic and will increase the difficulty of interaction with the menu, thus makes the design from an ideal paradigm to a frailer one. How to efficiently rotate the menu is a problem. There are front, back, left and right face on the cube. If the cube is rotated horizontally, the difficulty of interaction would decrease. And we come up with a semi-automatic navigational rotation system that controls the cube rotate around Y-axis horizontally to solve the efficiency problem. The following paragraphs presents the theory of this method.
2.2 3D Cube Menu

In this method, a user holds the PHANToM and makes it move. There is a button called "mainbutton" on PHANToM device, it passes the state to the rotation system to decide whether the menu is going to rotate. If the state shows the mainbutton is pressed, that means the user want to rotate the menu. The cube menu automatically rotates ninety degrees each time when it receives a signal, this signal tells the cube which direction it should rotate. If the state shows the mainbutton is released, then the system receives the signal that the user stops rotating the menu.

Further consideration is on the rotation direction. The rotation direction is estimated based on PHANToM’s movement. While direction and continuation are the two factors that are essentially considered on PHANToM’s movement, which means the system continuously tracks the PHANToM’s position to decide the rotation direction and how many times the cube rotates. If the system detects the PHANToM moves to the left, the menu will rotate ninety degrees to the left. If detects to the right, the menu will rotate ninety degrees to the right. Figure 2.3 shows the cube rotate ninety degrees to the right. It is quite common for such situation, the cube could continuously rotate ninety degrees for the same direction or different directions without discontinuation, if the main button is not released and PHANToM is always moving. There is a short time pause between each rotation in the continuous movement to ensure the users know what is on each face.

The smoothness effort made to the rotation is to simulate the continuous motion of rotation instead of rotate the cube in a frame rate. Since the cube only rotates on Y axis, after each rotation, the cube will be at the same position as before. The only difference is a another face is facing the user.

No mistake supposes to be made during the whole process, since the system accurately estimates what the user intention and makes execution.

Figure 2.3. A rotation animation ninety degrees to the right
2.2.2 Accuracy test

An accuracy test is designed in order to see how accurate the potential users interact with the menu in the aspect of option selection interaction. It detects how accurate the user presses the buttons. The results is studied to evaluate and analysis in normal conditions, with proper size of the menu, how good the users manipulate it. It is one of the measurements for how valuable the option selection method on the cube menu is for future research and practical uses.

Here’s the formula for accuracy test:

\[ Q_{\text{rate}} = \frac{N_{\text{real}}}{N_{\text{total}}} \]

Where
- \( N_{\text{real}} \) is the times of "real" buttons are pressed.
- \( N_{\text{total}} \) is the total times of all the buttons are pressed.
- \( Q_{\text{rate}} \) is the accuracy rate.

The idea of the accuracy test is, on certain faces, some buttons will be designed as decoys, and those buttons are around the "real" buttons. The accuracy rate is derived from the division between the total number of all the buttons that are pressed and the number of the 'real' buttons that are pressed.
Chapter 3

Implementation

Earlier chapters have described the background and the theory basis of the project. This chapter provides an overview of the functionality of these two menus, and details of the process, structure and realization for building the 2D pie menu and 3D cube menu.

3.1 The Implementation Overview

The 2D pie menu and 3D cube menu are implemented on H3DAPI platform with UI toolkit. H3D is a cross-platform scene-graph API and designed mainly to support a special rapid development process in the area of 3D graphics, haptics and sound. The structure of the API is based on X3D standard. The X3D standard is a XML based scene-graph and Virtual Reality definition file format. X3D can also load and export scenes. To implement advanced behavior, scripting language Python is applied. For both of the menus, menu framework is built by using UI toolkit for haptic 3D-widgets in H3DAPI. This UI toolkit contains nodes for creating widgets, such as buttons and sliders, which can be positioned in 3D space. The widgets are activated with a haptics device or a mouse. The existing UI nodes are Frame, GridInfo, GridLayoutManager, Label, PopupMenu, Sliderbar, TouchButton. Among them, Frame, PopupMenu, GridInfo, TouchButton, Label nodes are applied in our application. Frame node works similar to a grouping node, widgets can be placed in it. PopupMenu node is applied to pop up a submenu when pressing a button on pie menu. TouchButton node changes state when it is touched by the haptics device. For these two menus, if the user moves the PHANToM pen into the menu option of interest and press the button with PHANToM stylus, this button is triggered and the state of the button will become true. This mode is most similar to buttons used in most mouse-driven desktop systems. And in order to write text on the buttons, Label nodes are needed.

Section 3.2 and 3.3 elaborates the implementation method on pie menu and cube menu respectively. Pseudo code and graphs are included in some sections to clarify methods.
3.2 2D Pie Menu

This section describes the details on implementation of the pie menu. This menu is built by following steps:

1. Create a sphere as sample geometry.

2. Build the user interface framework, start with eight buttons on pie menu, and then a submenu which contains nine options for each button on pie menu.

3. Implement option functions of the menu, which includes scaling, diffuse color, shininess, specular color, texture and transparency. Build routes for transforming data between Python and X3D, and data in between X3D.

4. Add spring effect to each button on pie menu.

![Figure 3.1. Pie menu implementation process](image)

The functionality of the framework structure, spring force feedback are described in the following parts.

3.2.1 Structure

The pie menu is designed with eight buttons on the pie menu, and eight square submenus for those eight buttons on pie menu. The figure below shows the pie menu with scaling submenu pops up.

![Figure 3.2. Pie Menu](image)
To constitute a pie menu frame structure, there are eight PopupMenu nodes in the frame for eight buttons on the pie menu. Each PopupMenu node is constituted by a GridInfo node, a TouchButton node and a Frame node. The GridInfo node gives information about what position in the grid the widget is to be positioned in. The TouchButton node is a user interface button node. It changes state when it is touched by the haptics device. The Frame is a node for a frame in which widgets can be placed. In this case, nine TouchButton nodes are placed in the Frame node belonging to PopupMenu node. This frame is a submenu, and it pops up when the state of TouchButton in that PopupMenu node shows it is triggered. A tree structure of how the pie menu frame is built is showed in figure 3.3.

Figure 3.3. Tree Structure of Pie Menu

3.2.2 Menu category

There are eight buttons on the pie menu, of which six composed the menu category in this application. This category includes scaling, diffuse color, shininess, specular color, texture and transparency. Scaling, diffuse color, shininess, specular color,
and transparency have been initialized when creating the sample sphere.

Secondary options are in the submenus. When a button on the submenus is
pressed, corresponding change on the sphere will be made. For example, when
the scaling button is pressed, a square submenu with nine options for different
scaling size is on the right side of pie menu. If an option, such as a button with
text 0.02 in the submenu is chosen, the radius of the sphere will become 0.02. Diffuse colors, shininess, specular color, transparency selections act in the same
way. Similar method is used on texture application, but one more thing needed is
to create a function in Python to stores a MultiTexture node that containing all
the texture nodes for all texture buttons. When a texture button is triggered, the
Corresponding texture is mapped to the sphere.

3.2.3 Spring effect

The spring effect is implemented based on the theory described in the previous
chapter. When the PHANToM stylus is on the edge boundary, the force on the
PHANToM stylus reaches its biggest and forces the stylus moving towards the
button. This force is smaller and smaller as the stylus gets closer and closer to the
button. When the stylus is in the center of button plane, this force becomes 0.

There are three different ways to define edge boundaries. One is the cube edge
boundary method, and the other two are spherical edge boundary methods. In the
cube edge boundary method, and one of the spherical edge boundary methods,
such formulas are applied concerning on each button on pie menu:

$$\vec{V} = \vec{P}_{\text{position}} - \vec{P}_{\text{buttonposition}}$$

Where

- $\vec{P}_{\text{position}}$ is the position of the PHANToM that is updated in realtime;
- $\vec{P}_{\text{buttonposition}}$ is the button’s center position that is stable and known already;
- $\vec{V}$ is the distance between the PHANToM and the button’s center.

$$\vec{F} = - K_{\text{stiffness}} \vec{V}$$

Where

- $K_{\text{stiffness}}$ is the coefficient for stiffness;
- $\vec{F}$ is the force applied by the spring to the haptics device.

In the cube edge method, the stiffness constant is set to be 25, the edge bound-
ary is set to be 0.01 to the button’s center in $x$ and $y$ direction, and 0.1 in $z$
direction. While in the spherical edge method, the stiffness constant is also set to
be 25, distance between sphere edge and the button’s center is 0.01.

The other spherical edge method applies a SpringEffect node to each button
on the main menu. SpringEffect is a localized haptic effect where the haptics
device is pulled towards to a point in space in a spring like manner. So by using
the SpringEffect nodes, the same effect as the spherical edge boundary method
describes above is achieved.

In the real application, only the first cube edge boundary method is applied,
since the button is cubic, cube edge is more natural than spherical edge.
3.3 3D Cube Menu

This section describes the details of implementing the cube menu. This cube menu has four faces. They are left, right, front and back faces on the cube. Some are with functionalities, and some are without. Users manipulate the cube with a semi-automatic rotating system. The process of building up the 3D cube menu is as follows:

1. Build the user interface framework. Four faces are separately placed on the front, back, left and right face to form a cube. Nine buttons are created in each menu frame.

2. Implement functionalities on cube that includes loading geometries, textures and models. Build routes for transforming data between Python and X3D, and data in between X3D for those functionalities.

3. Implement semi-automatic rotating animation system on the cube.

4. Implement the functionality of accuracy test.

![Figure 3.4. Cube menu implementation process](image)

In the following parts, details on implementation of structure, menu category, rotation and accuracy test of the cube menu are described.

3.3.1 Structure

The 3D cube menu is composed by front face, back face, left face and right face. Nine buttons are placed on each face, some are real and can trigger events, while some are decoys and cannot trigger any event. Figure 3.5 shows what the cube menu looks like.

The framework structure of 3D cube menu is constructed in this way. Firstly, a main Transform node is built for rotating the whole cube. Then four Transform nodes are built hierarchically under the main Transform node. Under each hierarchical Transform node, there is a Frame node building up a face. Each hierarchical Transform node translates the belonging face to specific position to form a cube in horizontal plane. Nine buttons are placed in each face. Figure 3.6 shows the tree structure map for this cube menu.

3.3.2 Menu category

This part describes how to apply the menu category to the 3D cube menu.
The menu category includes "Geometries", "Textures" and "Models" categories on three faces respectively. The implementation method is similar to the method that is described in the "Menu category" part of the "2D Pie Menu" section in this chapter, that is when a button on cube is triggered, corresponding change will be made. One thing has to be pointed out, when users make selections between Geometries and Models in the menu, the system has to clean the geometry field to set a new model, or to clear the list storing the model for setting a new geometry in Python.

### 3.3.3 Rotation

The semi-automatic rotation animation system is simulated based on the rotation theory that is narrated in last chapter. The cube menu is manipulated to rotate horizontally around Y-axis either to the left or to the right. The decision of rotation direction is based on PHANToM’s movement.

The "mainButton" and the 'trackerPosition' are two fields of the haptics device,
which refers to the PHANToM. Previous one is the state of the stylus, and the later one is the stylus’ position in the virtual world. The semi-automatic rotation algorithm is simulated as follows. An update() function in Rotate class in the Python script is built, the update function will always execute as soon as an event is received. The mainButton field and the trackerPosition field of the PHANToM are imported from X3D to the Rotate class and are updated per frame rate. When the button on the PHANToM is triggered, the state of mainButton is true. Based on the changes on trackerPosition and the state of the mainButton, the system will decide whether to rotate and further estimate in which direction the cube rotate.

Once the button on the PHANToM is pressed, the updated "trackerposition" which refers to PHANToM’s position is inserted at the front of the list P, which is a list used to contain the PHANToM’s positions. So, the latest position is always at the front of P. If there are at least two elements in P, A temporary direction of PHANToM’s movement will be estimated. Such estimation is based on comparison between the first two elements in P. If the first element is bigger than the second element, pass 1 to the end of a list and name it Q. If the first one is smaller, pass 0 to the end of the list Q. When there are two elements in Q, make comparison of these two elements and the first element of Q is deleted. If those two elements are the same, and both equal to 1, that means the PHANToM tend to move the cube to the right. And then, accumulate 1 to markb, which is a marker. If they both equal to 0, that means the PHANToM tend to move the cube to the left, so accumulate 1 to marka, which is another marker. If the two elements are not the same, it means the PHANToM change the direction. Elements in P and Q will be cleared, and marka and markb are set to 0.

When a new PHANToM’s position is updated, the process described above will run once again. That is, pass the trackerPosition value to P, and make comparison of the positions, store the results and then compare these results to orient. If the results are continuously the same for certain amount of frames, that means the user instruct the cube to rotate, and the rotation animation begins. Once the results are not the same, elements in P and Q will be deleted, and mark values "marka" and "markb" are set to 0. The system will start all over again to calculate and estimate rotation. Calculation and estimation are based on the mainButton’s state and the updated trackerPosition of the PHANToM. The process will not stop until the mainbutton’s state is false.

The ToggleGroup is a node enable/disable haptics and graphics rendering of the cube menu. When the cube receives the signal to rotate, the ToggleGroup node disables the haptics rendering on the cube to escape the distraction from the PHANToM’s movement. Then the TimeSensor node combined with the OrientationInterpolator node simulates the rotation animation. The TimeSensor node generates events as time passes, it can be used in driving continuous simulations and animations, controlling periodic activities and initiating single occurrence events such as an alarm clock. In the node OrientationInterpolator, the key field contains list of key times, the keyValue field contains values for the target field, one complete set of values for each key, and in this case, the target field is the Rotation field in the main Transform node.

In the case of right rotation, it could appear as:
key[0 0.3 0.6 0.9]

keyValue[0 1 0 b.a 0 1 0 b.a+0.52 0 1 0 b.a+1.04 0 1 0 b.a+1.57]

b is the value for the Rotation field updated before the rotation animation, b.a is the axis of that rotation value, each set of rotation value in field keyValue is corresponding to each value in key field.

The current time is sent to startTime field in the TimeSensor node, so the rotation animation is started. And the haptics rendering is enabling to working. Elements in P and Q will be deleted, and mark values are set to 0.

New rotation animation will be re-estimated based on the state of mainButton and calculations on updated PHANToM positions. The figure 3.7 shows the structure of the 3D cube menu navigation and rotation animation.

![Diagram of 3D cube menu navigation and rotation](image)

**Figure 3.7.** Tree Structure of navigation and rotation

Here is the Pseudo-code of the whole navigation and rotation algorithm:

```pseudo
if main button is pressed
    then insert trackerposition at the front of the list P
if length(P) > 1
    then new <- P[0]
    sec <- P[1]
    if new.x - sec.x > 0
        then add 1 to the end of the list Q
    if new.x - sec.x < 0
        then add 0 to the end of the list Q
if length(Q) > 1
    then prev <- Q[0]
    delete Q[0]
    if prev == Q[0] and prev == 1
        then markb <- markb+1
```
if markb > 8
then set the cube group not be rendered haptically
if isActive false event is generated on TimeSensor
then do rotation animation
set the cube group not be rendered haptically
delete P
delete Q
markb <- 0

else if prev == Q[0] and prev == 0
then marka <- marka+1
if marka > 8
then set the cube group not be rendered haptically
if isActive false event is generated on TimeSensor
then do rotation animation
set the cube group not be rendered haptically
delete P
delete Q
marka <- 0
else
then markb <- 0
marka <- 0
delete P
delete Q

3.3.4 Accuracy test

As the accuracy rate theory stated in last chapter, the accuracy rate is derived from the division between the number of times the 'real' buttons are pressed and the number of times all buttons are pressed.

In the implementation, all the buttons' state on the cube menu that is involved in the accuracy test is updated in real-time. Once a button is pressed, if it is a real button, the variable storing the number of times the real buttons being pressed is accumulated by 1; if it is a decoy, the variable storing the number of times the decoys being pressed is accumulated by 1. The sum of these two variables becomes the total number of times all the buttons being pressed. Thus, the result of how accurate the button is being pressed is calculated and updated each time after a button on the tested faces is pressed.
Chapter 4

A Pilot Study

In this part, a pilot study is presented to assess the feasibility of the two menus, which are the 2D pie menu and the 3D cube menu that work in Virtual Reality with haptics and head tracking interaction. And the pilot study is also presented to get deeper insight of how these menus could perform better, how users of an application could have fine controls through the user interface to interact with the virtual environment. To work out the pilot study and write it down, we referenced the method from K.johan et al [11].

The main consideration of user interface is usability. Usability is the degree to which the design of a particular user interface takes into account the human psychology and physiology of the users, and makes the process of using the system effective, efficient and satisfying [22]. It also takes into account the requirements from its context of use. The most important purpose to work out a user study case is to find out how potential users react to these two menus, the menus’ performance during the interaction, and the users’ experiences of these menus, therefore analyze and evaluate the usability and feasibility of these two menus. Another purpose is the evaluation results are a key important consideration for further improvement on the user interfaces.

A list of tasks for each menu is executed in a story telling way, and a questionnaire is made for each menu. The questions in the questionnaire are designed based on three components of usability, which are learnability, efficiency and user’s satisfaction. Those questions are answered based on users’ experience by how much users agree with the statement. Then a discussion is made based on the performance of user interfaces and user’s experiences, the topic might not be covered by the questionnaire since some issues could not simply be graded agree or disagree, such as the overall feeling that are concerned on the menu, and the improvement that is expected to be made. Therefore at last, we make evaluations and discussion based on the work described above and give ideas for future improvement.
4.1 Software

The case is set for users to better experience how these menus perform, thus the supervisor can make proper evaluation based on users’ reactions and feedback. The case is basically taken from geometries. For the pie menu, the tasks in current case are to set different properties on the geometry. And for the cube menu, the tasks are set to view different geometries, models and textures. Task completion time for each user on each menu is collected as a measurement for efficiency.

In the pie menu, the geometry for the case is a sphere initialized with radius 0.023, in gray color, positioned on the right side of the menu. There are eight buttons on the pie menu, of which six are used to set different properties of the sphere. Those properties include scaling, diffuse color, shininess and specular color for highlight, texture and transparency.

The descriptions for the properties are as follows:

**Scaling** It scales the radius to different sizes.

**Diffuse color** It reflects all light sources depending on the angle of the surface with respect to the light source. The more directly the surface faces the light, the more diffuse the light reflects.

**Shininess** It affects the specular highlights. Lower shininess values produce soft glows, while higher values result in sharper, smaller highlights.

**Specular color** It is the color of the specular highlights.

**Texture** It sets textures for the sphere. The number of pixels is different between various textures, typically with $720 \times 360$ pixels or more.

**Transparency** It sets how clear the sphere is, with 1.0 being completely transparent and 0.0 completely opaque.

For the 3D cube menu, the case is make selections between geometries, textures on geometries and models. There is one face containing different geometries, another face containing the textures that are mapped on the geometries, and a third face containing various x3d models. There’s one face left in the menu with all buttons, but no function is implemented on those buttons. Users might take notice of what is the difference between this face and the other three faces. An accuracy test is taken on the face "Geometries" and face "Models". The decoy buttons are placed between the real buttons, and the real buttons are different geometry or model options. The accuracy rate the user presses those real buttons is calculated, the result is a measurement of how good the option selection performed without any other assistive effects.

4.2 Procedure

There are six people participated in test, five have previous experiences working with haptics in virtual environment, of which two have experiences of headtracking
in virtual environment. Each participant is invited to a private supervised session for the following steps:

1. Introduction to the pie menu.
2. Exploration of the pie menu.
3. Introduction to the 3D cube menu.
4. Exploration of the 3D cube menu.
5. Informal interview, questionnaire and discussion.

Though no specific training data is set for users to familiarize each menu before each introduction, but there is a small trial between each introduction and exploration, users are informed to try to touch the buttons to familiar with the interface and the spring force around each button on pie menu, and try to get used to the rotation animation instead of conventional mode. To rotation the cube animatedly 90 degrees to the left or right, users slide the PHANToM stylus to the left or right and at the same time press the mainbutton on the stylus.

During each exploration, each user is asked to express the feeling of the interaction with these menus. It is more of like a conversation between the user and supervisor. An accuracy test, which refers to how many times the "real" buttons are pressed compared to how many times the decoys are pressed, is taken for each user when doing tasks on the cube menu. After that, each user answers a questionnaire regards to the experiences of both menus, in a way of how much did the user agree or disagree with each statement in the questionnaire. At last, a discussion is made to discuss the drawbacks, advantages and experiences that are not covered in the previous talk and questionnaire. The completion time is counted for each exploration on each user as a measurement and evaluation basis.

4.3 Results

The assessment of the pilot study is a formative assessment, it typically involves qualitative feedback. For example, how the user experiences the menus, how the user feels spring force feedback help option selection, how the rotation makes the menu more intuitive, and the preferences of the users. Conversation, questionnaire and discussion are all based on user interface design, users’ exploration, and users themselves’ experiences. And those answers, feedback and opinions are the main sources applied in this part. The result of questionnaire is in appendix A.

In the exploration of pie menu, all the users gave the feedback that it is easy to learn how to use the menu, although one user who had no experience with Virtual Reality before had to take long time to familiar with the interaction. Almost all users totally agreed that it is straightforward in option searching. Half of the users did not agree that it is hard to change from one button to another under the help of assistive forces. All agreed or tended to agree that the PHANToM stylus is attracted by the buttons. And five users agreed that the force suddenly shows up
when the stylus is near the button could be annoying, the other one tended to not agree with this point, one of those five users mentioned that the force is too big at the border. Half of all users agreed that the assistive force might lead the pen to neighboring buttons, the other half fully or to some degree did not agree with the term. Related to this issue, one mentioned that the stylus is very struggling between two buttons, and another one thought the force might need to be more accurate, it could lead the stylus to another button that is not the target. Also, it could meet the following case, if the user wants to press one button, but he changes his mind and wants to choose another one, it would be very difficult to change the track. Five did not agree that the submenu adds complexity to the interaction, and one of them expressed that she really likes the submenu since it clearly shows the category on main menu, and specific submenu gives more choices regarding to the corresponding item. But the user without Virtual Reality experiences before felt the submenu makes the whole system complicated. All agreed that the submenu increases the sense of item grouping. Four users fully agreed and the other two tended to agree that the system reacts as they expected.

One user mentioned that the change on the buttons before and after selection is indistinct. It is hard to notice which button is triggered on submenus. Another one said that he was confused with which button each submenu belongs to, since no clear sign shows it, and the highlight of that button is not distinct enough.

In the exploration of 3D cube menu, five users agreed that it is easy to learn how to rotate the cube, and one did not agree. All users agreed that is easy to navigate the rotation and the rotation animation works smoothly. Four felt difficult to see what it is on the back face, two did not or partially did not agree with this term. Two of the participants mentioned that it would be better to add a preview for the menu. Three fully agreed that it is easy to press the target button, one tended to agree and two was neutral. There was one user mentioned that she did not know how much force should be applied to press the button successful. Four and two respectively agreed or tended to agree that the operation of rotation is intuitive, although one mentioned that it could be hard to start using rotate system without instructions. Five agreed that the rotation is flexible, which means no need to push or pull the cube. Only two noticed the text color on buttons is changed when the button is pressed, which is out of the supervisor’s expectation. Four fully and two to some degree agreed that the menu reacts as they expected. One user tried to rotate the menu in a way the stylus pushes the cube. In this case, he might accidentally press some buttons when the rotation is about to finish even though he did not intend to press them, and he expressed that he would more prefer the conventional manipulation mode, which means the stylus push the cube to move.

The assessment also covered quantitative feedback. The table for accuracy result on cube menu is showed in table 4.1. And the table for task completion time on each subject is showed in table 4.2. It is the same amount exploration tasks on pie menu and on cube menu, but the completion time on the pie menu is remarkably less than it on the cube menu for four users’ exploration, for about thirty seconds to one minute, while the other two participants finished them with almost the same time.
4.4 Discussion

<table>
<thead>
<tr>
<th>Accuracy Rate</th>
<th>Subject 1</th>
<th>Subject 2</th>
<th>Subject 3</th>
<th>Subject 4</th>
<th>Subject 5</th>
<th>Subject 6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cube Menu</td>
<td>0.875</td>
<td>0.750</td>
<td>0.889</td>
<td>0.875</td>
<td>0.833</td>
<td>0.652</td>
</tr>
</tbody>
</table>

Table 4.1. The accuracy rate for the cube menu

<table>
<thead>
<tr>
<th>Task Completion Time (min)</th>
<th>Subject 1</th>
<th>Subject 2</th>
<th>Subject 3</th>
<th>Subject 4</th>
<th>Subject 5</th>
<th>Subject 6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pie Menu</td>
<td>1.38</td>
<td>1.41</td>
<td>1.27</td>
<td>2.23</td>
<td>2.37</td>
<td>4.00</td>
</tr>
<tr>
<td>Cube Menu</td>
<td>2.30</td>
<td>2.30</td>
<td>1.98</td>
<td>3.44</td>
<td>2.43</td>
<td>5.12</td>
</tr>
</tbody>
</table>

Table 4.2. The task completion time for the pie menu and the cube menu

Generally speaking, it is easy to get started using those menus. Spring force feedback is considered helpful for option selection. Navigation and rotation animation system is efficient, accurate and intuitive.

4.4 Discussion

The discussion and analysis in this section are made based on sources including the users’ behavior and expression during exploration, questionnaire and discussion. The evaluation is provided as a demonstration and material for future development and further research.

Participants have expressed that both the 2D pie menu and 3D cube menu are in good control and intuitive during the interaction, and different participants have different preferences on the menus. Although few users felt a bit uncomfortable to control the spring force at the border or in the space between the neighboring buttons, and one user expressed that he might press buttons on the cube menu unconsciously when he tried to rotate the cube, it makes him feel annoyed. Actually, the force on the border could be adjusted to be either bigger or small to make users feel comfortable with while at the same time make the selection faster and more accurate. The reason for the buttons were pressed unconsciously is that he kept pushing the cube every time he wanted to rotate, it would not be a problem if he keeps the stylus some distance from the cube when managing to rotate. During and after the pilot study, we realized that if the instructions are more detailed or even more guiding before the use, users would face fewer problems and feel more freely to interact with these two menus.
Here comes the comparison between these two menus based on the result from the pilot study. It clearly shows from the data in table 4.2, for each of the six subjects, he/she took considerably more time to finish the tasks on the cube menu than the tasks on the pie menu. The comparison is made under the condition that two menus have the same amount of tasks. This may implicate that pie menu is more time efficient than the cube menu is. It has been shown in the paper from Rick and Colin [9], the pie shape and force feedback could lower the error rate, but from the accuracy test taken on the cube menu in this pilot study, all subjects got certain high accurate results except one who had no haptics experience before. This shows that even without force feedback on the buttons, normal quadrilateral shape with no force feedback on the buttons might be accurate enough to be used. Compare to the hierarchical structure on the pie menu, the non-hierarchical structure on the cube menu is rather simple. In this case, the pie menu can include more categories and more options. The pie menu and the cube menu are rather different types, one is 2D and the other is 3D, almost all expressed that the 3D one is more interesting and achieves better visual effect. But without navigation rotation system, the cube menu is much more difficult to interact with and much less usable, while the 2D pie menu will not have any problem with manipulating the menu.

Though we have made comparison between the pie menu and cube menu, it is really hard to conclude which one performs better than the other since they have their own characteristics. When designing the menu, there are a lot to consider, such as feasibility, difficulties, user preferences and so on. In the future design, the user interface might combine the elements in the pie menu and the cube menu to work out a desired model, for example a 3D cube menu with spring force feedback on the buttons.
Chapter 5

Conclusion and Future Work

5.1 Conclusion

Usually, the multi-model human machine interacted systems in Virtual Reality is not usable without user interfaces. This project is aim to provide good solutions to make human machine interactions more efficient. A multi-model user interface with a 2D pie menu and a 3D cube menu is implemented under H3D API with UI toolkit and simulated in VR workbench. H3D API is ideal for implementing multi-model applications that is based on 3D graphics, haptics and sound in Virtual Reality. While UI toolkit provides a good framework structure that could be used to build menus in different shapes, in different button modes and in popup style efficiently. In this work, X3D combined Python builds the user interfaces, defines geometries, arranges scene-graph elements, and makes good control of these two menus.

In all, this project has achieved its goal. The 2D pie menu with force feedback is a fast and low error rate approach with good control. The 3D cube menu is in a future promising user interface style. The system precisely detects the user’s purpose in which direction to rotate, how many continuous times the rotation animation to be made. The three dimensional structure and the rotation animation made the menu achieves a good visual effect.

Here list four aspects of how good the 2D pie menu have achieved:

**Efficiency** It is a rather fast and accurate approach, both "pie" shape and spring force could save time and increase accuracy. Four out of six participants completed the exploration on pie menu much faster than the exploration on cube menu. The buttons are dispersive placed in an order, and when the stylus is near a button in certain range, the system considers that button is of interest and automatically forces the stylus move towards to the button.

**Guiding** Unlike in the conventional systems, the user moves the mouse to the menu option of interest and release the button, this pie menu system applies spring force to force the stylus move towards to the option and push the
stylus into the plane of the menu, thus a selection is made. This directly guides the user to select the option of interest.

**Accuracy** It has been shown that spring force feedback on buttons can lower the error rate on option selections.

**Clarity** Participants have expressed and agreed that the menu clearly shows the content with item grouping. Since the structure of the menu system is simple and clear. The main menu is a catalog with eight options, and eight hierarchical menus belong to those eight options on pie menu respectively.

Here list five aspects of how good the 3D cube menu have achieved:

**Flexibility** Participants agreed that navigation and rotation increase the subject’s flexibility. The cube could rotate to the left or right for 90 degrees each time based the PHANToM’s button state and the stylus’ movement, there is always one face exactly faces towards the user.

**Intuitivity** The rotation direction depends on the PHANToM’s movement makes menu intuitive, although one participant expressed that the stylus may collide with cube when rotating. Besides the rotation, no submenu is on the cube, which also makes the whole menu intuitive.

**Maneuverability** Since the menu is three-dimensional, it is more complicated to control than the 2D menu. Instead of the stylus grabs the cube to rotate manually which is rather hard to control, fast and accurate rotation system manipulates the menu in good condition.

**Accuracy** The method applied provides accurate navigation. All the participants expressed the cube’s movement follows their intentions.

**Clarity** Participants agreed that the menu shows up the content clearly. Each face of the cube menu is a group of similar items with group name in the middle.

As stated, both of these two menus had performed well from users’ feedback. With their own characteristics, they are suitable for different situation respectively. Those situations include the number of options and hierarchy level the menu will be applied, how many space for the menu, how accurate for the option selection is expected and so on. And how good menus would perform is partially depended on different parameters in different situation. It is essential to find the most appropriate menu with proper parameters to most effectively manipulate the interactions in Virtual Reality. Inappropriate types of menu and parameters on menus will not help or even give negative impact to the interaction. Besides, different users might have different preference, some would more prefer menus with rotation animation, while the others would more prefer menus with spring force. In order to achieve desirable effect, at the beginning of the project, comprehensive consideration should cover the types of menu, parameters, intuitivity, manipulation, the preferences of potential users and etc.
5.2 Future Work

Based on self evaluation and results from the pilot study, this section lists several points that could be considered improving in the future.

For the pie menu, force feedback control could be developed better between neighboring buttons and on the force’s border. Changes on buttons before and after selection could be improved to be more distinct, so that users could notice what he selects even readily. Displaying what have been chosen is a good option, so the users can clearly realize what had been done.

For the cube menu, it could apply a display of menu preview to be with the cube menu to increase clarity, so that users could know which face he is facing and what are the left faces doing. Another idea is adding secondary cubes to the 3D cube menu system if more categories are needed since the number of categories menu is rather limited in the current case. One hypothesis is when an option on the main cube is released, the main cube shrinks till disappears, and then at the same position, the secondary cube menu of that option enlarges to the main cube’s original size.
Bibliography


Appendix A

Questionnaire

The questions in the questionnaire are answered with a value between one and five corresponding to "do not agree" and "fully agree", respectively (Likert scaling). Each question listed below is marked with a letter, corresponding to a line in Figure A.1.

1. The 3D cube menu:

   A ...is easy to learn how to rotate
   B ...is easy to rotate from side to side
   C ...the rotation animation is working smoothly when it rotates ninety degree
   D ...is difficult to see what it is on back face
   E ...is easy to choose the buttons that is wanted
   F ...the operation of rotation is intuitive
   G ...the rotation is flexible as the users do no need to push the cube
   H ...user noticed the text color is changed on the buttons
   I ...feels the system reacts as the user expected

2. The 2D pie menu:

   J ...is easy to learn how to use the menu
   K ...is straightforward in option searching
   L ...is hard to change from one button to another on pie menu
   M ...feel the buttons on pie menu work as a magnet to the pen
   N ...the force suddenly shows up when the pen is near the button could be annoying
   O ...assistive force might lead the pen to the neighboring buttons
   P ...the submenu added complexity to the interaction
Q ...the submenu increased the sense of item grouping
R ...feels the system reacts as the user expected

Figure A.1. The answers from the questionnaire. Each question is answered using a five level Likert scale. The order and colour of the blocks corresponds to the answer and their size, and the number in each block corresponds to the number of users giving that answer.