

Enhancing Precision for Object Manipulation in Virtual Reality Applications

LORENZ KRAUTGARTNER



MASTERARBEIT

eingereicht am
Fachhochschul-Masterstudiengang

Interactive Media

in Hagenberg

im September 2017

© Copyright 2017 Lorenz Krautgartner

This work is published under the conditions of the *Creative Commons License Attribution–NonCommercial–NoDerivatives* (CC BY-NC-ND)—see <http://creativecommons.org/licenses/by-nc-nd/3.0/>.

Declaration

I hereby declare and confirm that this thesis is entirely the result of my own original work. Where other sources of information have been used, they have been indicated as such and properly acknowledged. I further declare that this or similar work has not been submitted for credit elsewhere.

Hagenberg, September 25, 2017

Lorenz Krautgartner

Contents

Declaration	iii
Abstract	vii
Kurzfassung	viii
1 Introduction	1
1.1 Motivation	2
1.2 Objectives	2
1.3 Structure	2
2 Related Work	4
2.1 Manipulation	4
2.1.1 Canonical Manipulation Tasks	4
2.1.2 Application-Specific Manipulation Tasks	5
2.2 Input Devices	5
2.3 Degrees of Freedom	7
2.4 Widgets	10
3 Existing Techniques	12
3.1 Direct Manipulation	12
3.2 WIM, Worlds In Miniature	13
3.3 Go-Go Interaction Technique	13
3.4 HOMER	15
3.5 Voodoo Dolls	16
3.6 Proprioception-Based Interaction	17
3.7 Scaled-World Grab	18
3.8 SkeweR	19
3.9 PRISM	19
3.10 3-Point++	20
3.11 Handle Bar Metaphor	21
4 Interaction Design	22
4.1 Interaction Design for Immersive Virtual Environments	22

4.1.1	Stereoscopic vision	23
4.1.2	Intuitiveness vs Precision	23
4.1.3	Menu Interaction	23
4.2	Use Case	24
4.3	Included Features	24
4.3.1	Manipulation	25
4.3.2	Selection	28
4.3.3	System Controls	29
4.4	Design of PROMT	29
4.4.1	Overview	30
4.4.2	Input device	30
4.4.3	Widget	30
4.4.4	Axis Alignment	31
4.4.5	Translation	33
4.4.6	Rotation	33
4.4.7	Scale	33
4.4.8	Snapping Grid	34
4.4.9	Selection	35
4.4.10	System Controls	36
4.4.11	Scaled Input	37
5	Implementation	39
5.1	Technical Setup	39
5.1.1	Development Environment	39
5.1.2	Virtual Reality Setup	40
5.2	Application Architecture	40
5.3	Main Parts	41
5.3.1	Primitives	41
5.3.2	Widgets	42
5.3.3	Controller	43
5.3.4	Tool Manager	43
5.3.5	PROMT Tool	44
5.3.6	Managers	44
6	Evaluation	46
6.1	Methodology	46
6.2	Setting	47
6.3	Results	48
6.3.1	Hardware Interface	48
6.3.2	Manipulation Metaphors	48
6.3.3	Axis Alignment	49
6.3.4	Selection	49
6.3.5	System Controls	49
6.3.6	Grid Snapping	49

Contents	vi
6.4 Analysis	50
7 Conclusion	52
7.1 Summary	52
7.2 Future Work	53
A Protocol from the Expert Interview	55
A.1 Interview 1	55
A.2 Interview 2	64
A.3 Interview 3	70
B Contents of the CD-ROM	80
B.1 PDF files	80
B.2 Project Executables	80
B.3 Other	80
References	81
Literature	81
Online sources	83

Abstract

The advances in virtual reality system and their increasingly important role on the consumer market has led to a rapid development in the area of immersive virtual environments. This calls for an improvement in the field of human-computer interaction in virtual reality applications. Especially interaction design is an important factor that is still rapidly changing. Applications that focus on performance rather than playful interaction during object manipulation are still rare. Modeling and design software used for industrial purposes could benefit from the advanced perception in immersive virtual environments. To advance the research towards more precise object manipulation, this thesis will explore the design and implementation from PROMT, a precise, remote object manipulation technique. The technique focuses on combining and enhancing existing techniques to create a new interaction design for controller-based object manipulation based on gestural input. The goal of this thesis is to discover whether PROMT and other existing techniques used in virtual environments are potentially capable of providing users with the precision needed for industrial applications.

Kurzfassung

Die Fortschritte von Virtual Reality Systemen und ihre immer wichtiger werdende Rolle am Verbrauchermarkt haben zu einer schnellen Entwicklung im Bereich immersiver virtueller Umgebungen geführt. Dadurch wird eine Verbesserung im Bereich der Mensch-Maschine-Interaktion für Virtual Reality Anwendungen notwendig. Besonders Interaktionsdesign ist ein wichtiger Faktor, der sich immer noch schnell ändert. Anwendungen, welche sich mehr auf Leistung als auf spielerische Interaktion während der Manipulation von Objekten konzentrieren, sind immer noch eine Ausnahme. Software zum modellieren und entwerfen von Objekten für industrielle Anwendungen könnten von der besseren Wahrnehmung in immersiven virtuellen Umgebungen profitieren. Um die Forschung hin zu präziser Manipulation von Objekten weiter zu führen, befasst sich diese Arbeit mit der Entwicklung und Implementierung von PROMT, einer Technik zur präzisen Manipulation von entfernten Objekten. Diese Technik konzentriert sich darauf bestehende Techniken zu kombinieren und zu verbessern, um ein neues Interaktionsdesign für controller-basierte Objektmanipulation basierend auf Gestiken zu entwickeln. Das Ziel dieser Arbeit ist herauszufinden, ob PROMT und andere bereits existierende Techniken für virtuelle Umgebungen potenziell dazu in der Lage sind, BenutzerInnen die nötige Präzision für industrielle Anwendungen zu bieten.

Chapter 1

Introduction

There is a large number of computer applications that require some kind of three-dimensional object manipulation. Some examples are applications used for three-dimensional modeling, sculpting, virtual exploration, engineering, architectural visualizations and assembly. Because of this, the manipulation of three-dimensional models has always been an interesting problem, as it was mainly performed through the two-dimensional view of a computer screen. Therefore a lot of metaphors had to be found and established in terms of three-dimensional interaction on a two-dimensional screen. Also input devices such as the mouse and keyboard only enabled a two-dimensional input.

While this seems counter-intuitive, numerous possibilities have been found to overcome such mapping problems. Stephanie Houde [9] for instance developed the idea of a bounding box manipulated by handles which is, as seen in figure 1.1, very reminiscent of the widget approach that has become best practice for three-dimensional object manipulation.

However, the new availability of head-mounted stereoscopic displays and hand-held controllers that are tracked in 3D space enables a more native approach to three-dimensional object manipulation. The intuitive interaction of simply grabbing an object and moving it with the hand is easy to learn and fast. It also allows manipulation with six degrees of freedom, but it lacks the accuracy that is necessary in professional work.

In this thesis, existing interaction techniques for object manipulation will be discussed. The focus will be on interaction techniques for head-mounted displays or HMDs in combination with controllers. Also the development of a new technique called PROMT or Precise Remote Object Manipulation Technique will be presented and evaluated.

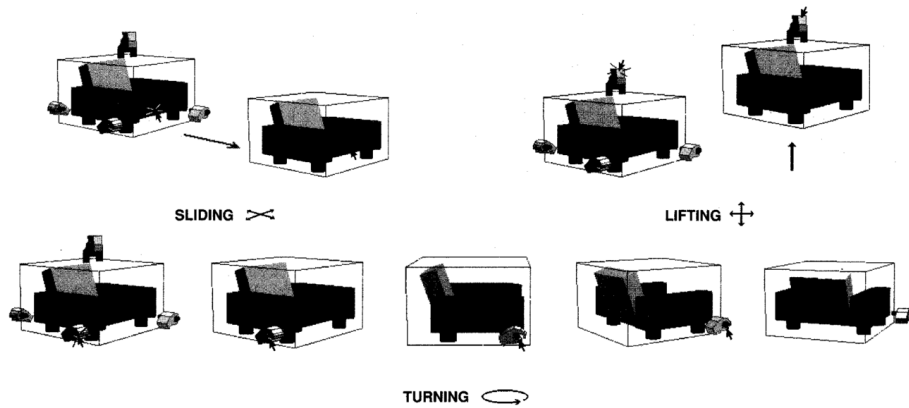


Figure 1.1: Visualization of the handle box approach by Houde [9].

1.1 Motivation

When working on immersive virtual environments, a lot of problems must be solved which often have few to no existing solutions. One interesting problem is the absence of best practice approaches in terms of user interaction. This becomes increasingly obvious when not thinking about playful interaction but industrial applications. Even problems as basic as manipulating three-dimensional objects still have no apparent solution, other than assimilating interaction techniques used in two-dimensional applications and directly interacting with them using controllers with six degrees of freedom. Therefore the focus of this thesis is set around this problem, to advance the research of user interaction in immersive virtual environments.

1.2 Objectives

The goal of this thesis is first and foremost to create a controller-based interaction technique for immersive virtual environments which improves the user's precision compared to existing techniques while maintaining speed and ease of use. This technique relies on knowledge gained from existing techniques of mid-air manipulation as well as desktop-based manipulation. Additionally a use-case application is designed which contains said technique and is later used to evaluate its feasibility in expert interviews.

1.3 Structure

This thesis is structured in seven chapters. Chapters 2 and 3 introduce existing research in the field of object manipulation in immersive virtual environ-

ments and grant an overview over techniques that already exist. Chapter 4 shows the development of the technique's design, as well as the design of the use case that acts as a proof of concept for the technique. This is followed by a description of the final implementation of the application in chapter 5. The aspects of the application relevant for PROMT are then evaluated in chapter 6 followed by a conclusion showing findings and potential future work.

Chapter 2

Related Work

This chapter provides an overview of related work for object manipulation in immersive virtual environments. It includes information in the fields of three-dimensional input devices, manipulation tasks, and general interaction design for immersive virtual environments. As this thesis covers the design of an interaction technique for virtual environments, focusing on precision and speed during object manipulation, these following sections have a great impact on the design decisions made in later chapters.

2.1 Manipulation

Manipulation typically describes the act of handling physical objects with one or two hands. Three-dimensional manipulation has a very similar goal. This goal is to move an object in three-dimensional space, for instance within an immersive virtual environment. Finding an appropriate three-dimensional manipulation technique can only be done by considering a certain manipulation task. Moving virtual objects in an immersive modeling application will need a vastly different technique than handling surgical instruments in a medical simulation. In this thesis, manipulation is defined as the tasks of spatial manipulation of rigid objects. Bowman et al. [3, p. 267] have separated these manipulation task in canonical manipulation tasks and application-specific manipulation tasks.

2.1.1 Canonical Manipulation Tasks

Canonical Manipulation Tasks describe the very basic manipulation tasks that make up more complex manipulation tasks. This makes it possible to focus the design and evaluation of techniques on a small subset of basic tasks, rather than considering all the more complex ones. Therefore Bowman et al. [3, p. 267] have designated four basic tasks as canonical manipulation tasks, which correspond with the tasks considered for the development of

PROMT. The tasks are based on physical manipulation of objects performed in the real world including reaching/grabbing, moving, and orienting objects. Additionally the very unnatural interaction of making objects bigger and smaller is considered because of its significance in three-dimensional design applications.

- **Selection** describes the task of choosing a particular object or acquiring it. This can be translated into the real world as picking up an object or pointing at it. The selection technique is the first step of object manipulation and influences the following manipulations.
- **Translating** an object is the task of changing its position. This means moving it from a starting position to target position.
- **Rotation** describes changing an object's orientation around a certain point in space; rotating it from a starting orientation to a target orientation.
- **Scaling** is the only task that has no proper real-world counterpart. It describes the task of changing an object's size. This is not a natural interaction, however a very common one in two-dimensional and three-dimensional object manipulation.

For each of these canonical tasks, there are also variables that affect user performance and usability. For instance the distance between a user and an object influences how well selection techniques perform. Also the size of the object and the amount of manipulation, e.g. how far an object has to be moved, and the required precision, are parameters to be considered.

2.1.2 Application-Specific Manipulation Tasks

Contrary to canonical manipulation tasks, which simplify manipulation tasks to their basic properties, application-specific manipulation tasks are aimed at a very specific form of manipulation. This may result in a manipulation task capturing aspects of a certain interaction that is made to fit only for a certain task. This could be tasks like moving the control stick of a virtual airplane. Because of their very specific nature, application-specific tasks are hardly generalizable.

2.2 Input Devices

There is a strong relation between interaction techniques and input devices. The properties of input devices used to capture the user's input has a significant impact on how interaction techniques are designed. The reason for this relation is that certain device features can restrict which interaction techniques can be used.

There have been a number of unique solutions for input devices in immersive virtual environments throughout their history. They can differ greatly

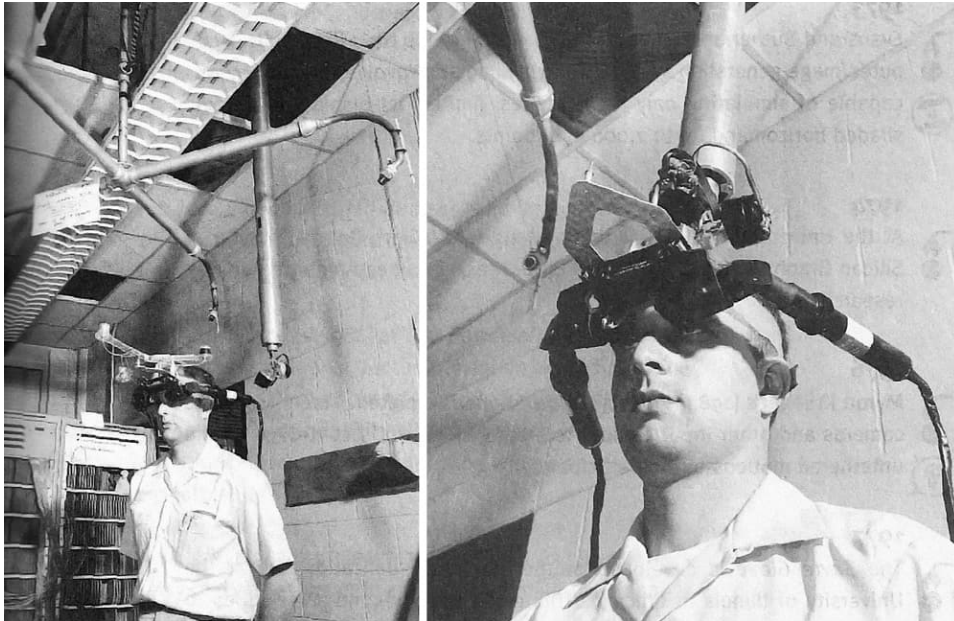


Figure 2.1: *Sword of Damocles* head-mounted display developed by Ivan Sutherland [20].

in the way they sense movement, including technologies using magnetic sensing, mechanical sensing, inertial sensing, optical sensing, and radar sensing. An example for early tracking of a head-mounted display is the *Sword of Damocles* (figure 2.1) developed by Ivan Sutherland [20] which allows for mechanical or magnetic tracking of the user's head. Current consumer systems mostly rely on optical sensing and inertial sensing. Another key factor is whether a system tracks the position of a controller or its movement. Due to the fact that systems tracking only the controllers movement can become offset and need calibration, a positional tracking is mostly preferred.

Other than the tracking technology also the number of control dimensions, e.g. how many degrees of freedom the device can control, and how they are integrated are an important factor in choosing an input device. A traditional mouse allows for integrated control with two degrees of freedom. Virtual reality controllers like the *HTC Vive*¹ controllers allow for integrated control with six degrees of freedom, tracking the controller's position in three-dimensional space as well as its orientation. Typical game controllers on the other hand provide four degrees of freedom, however those are separated, giving the user two degrees of freedom located on two joysticks. Zhai Shumin and John Sanders [23] have found that integrated control is usually

¹<https://www.vive.com>, last visited 13.09.2017



Figure 2.2: *CyberGlove Model III* data glove by *CyberGlove Systems*.

preferable because it allows the user to interact with an input device more naturally.

Spatial input devices, which utilize the position of a user's hand, head, or a single finger, are very often needed for three-dimensional applications in virtual environments. Those devices can often attempt to capture the movement of the human hand as closely as possible, for instance with data gloves (see figure 2.2)² or the *Leap Motion* system (see figure 2.3)³.

But these often lack reliability or are not able to capture the hand's movement closely enough, resulting in an uncanny interaction. Using an approach that is more abstract, like utilizing some form of controller that does only track the controller's position, has enabled more reliable input, especially in combination with more reliable tracking technology. One example for this is the *Cubic Mouse* developed by Fröhlich et al. [8] (see figure 2.4).

2.3 Degrees of Freedom

A degree of freedom describes a body's specific and independent movement in space. An input device like a computer mouse has two degrees of freedom

²A data glove is an input device in form of a glove.

³<https://www.leapmotion.com/>, last visited 04.09.2017



Figure 2.3: *Leap Motion* controller.

to move in, the x - and y -axes of a flat surface. A tracker, as found in head-mounted displays and tracked controllers, can sense its position in three axes and its orientation in three axes. Therefore such a tracker facilitates input in six degrees of freedom. The degrees of freedom of an input device can also be separated per device. A common gaming controller allows for four degrees of freedom separated into two sticks, where each stick allows for movement in the x - and y -axes.

In traditional mouse-based interaction systems, as well as in touch-based systems, the separation of the degrees of freedom for manipulating an object in a virtual environment has led to improved user performance when compared to direct approaches. This is mainly due to the mapping of a three-dimensional environment to a two-dimensional screen, which is required for such systems. In three-dimensional interaction systems, this mapping is not necessary as input and output happen in the same space. Nevertheless, the separation of the degrees of freedom in immersive virtual environments can be beneficial to increase precision.



Figure 2.4: *Cubic Mouse* controller by Fröhlich et al.

Mendes et al. [11] have conducted a user evaluation to determine whether the positive impact of degrees of freedom separation in mouse and touch interaction can also be seen in three-dimensional object manipulation within immersive virtual environments. In this evaluation, Mendes et al. compared three different techniques. The first technique was the direct manipulation technique described in section 3.1, which allows users to directly grab and move an object according to the movement of the user's hand. The second technique was the PRISM technique, as described in section 3.9, which was designed to improve precision during direct manipulation by scaling the user's hand movement according to the movement speed of the user's hand. The third technique is the technique developed by Mendes et al. [11] that allows separation of degrees of freedom using a virtual widget. This technique's representation and functionality is similar to the one introduced by Conner et al. [4].

The results of this evaluation have shown that direct manipulation and widget-based degrees of freedom separation are similarly easy to use. Direct manipulation has however shown quicker results in more complex manipulation tasks, whereas widget-based degrees of freedom separation outperformed direct manipulation as well as PRISM in terms of precision. Mendes et al. [11] suggest that this increase in precision is due to the fact that

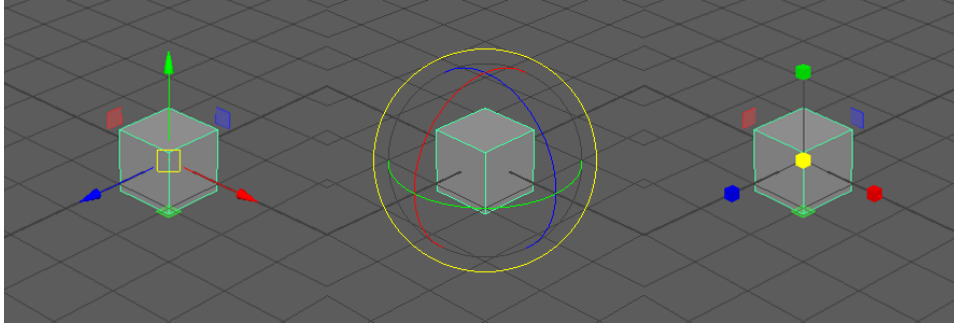


Figure 2.5: Manipulation widget used in the 3D modeling software *Autodesk Maya*. From left to right: rotation, scale, translation.

degrees-of-freedom separation prevents users from manipulating objects in an axis that they do not intend to manipulate.

2.4 Widgets

Conner et al. [4] defined a widget as:

[...] an encapsulation of geometry and behavior used to control or display information about application objects.

Widgets are forms of graphical user interfaces used to control an application. They are often used in mouse-based object manipulation tasks. One of the most prominent widgets is the three-dimensional manipulation widget used in many application for modelling or manipulating three-dimensional objects such as *Unity 3D*⁴ and *Autodesk Maya*⁵ (see figure 2.5).

This form of widget is context-sensitive; it is connected to the object that is being manipulated directly by position and orientation. This operation could also be done using some form of menu, but placing the interaction directly on the object is more intuitive as the functionality is directly connected to the object. This particular kind of three-dimensional widget is used to separate the degrees of freedom in which the user can manipulate an object, as described in section 2.3. By selecting the handles of the widget, the user intuitively selects an axis and sometimes also chooses whether to rotate, scale, or translate the object.

Based on this research, many techniques following different approaches have been developed over the last decades. The basic question of how techniques for selecting and manipulating objects should be designed has inspired many. The following chapter provides an overview over some of the

⁴<https://unity3d.com>, last visited 08.08.2017

⁵<https://www.autodesk.com/products/maya/overview>, last visited 08.08.2017

most important interaction techniques designed for object manipulation in immersive virtual environments.

Chapter 3

Existing Techniques

The field of user interaction in immersive virtual environments has so far brought numerous attempts of solving common problems in virtual reality interaction. These include such endeavors as finding techniques to select objects or extending a user's reach, as well as manipulating objects in different ways for different scenarios. Most of these techniques are very specialized for certain problems. In this chapter, the most influential and most popular techniques are examined individually, showing their strengths and weaknesses to give an overview of existing techniques which influence the development of PROMT. In this thesis only techniques that use some sort of hand-held controller or that could be used in conjunction with a hand-held controller are considered.

3.1 Direct Manipulation

Direct manipulation, also called simple virtual hand [3, p. 275], is an extremely intuitive way of interacting with an object. Direct manipulation in this context means that in an immersive virtual environment the position and orientation of a user's hand is used to directly manipulate a virtual object. An object can be selected by overlapping the virtual hand with the object, thus the user has to be close to the virtual object to select or grab it, as he or she would have to be when naturally interacting with an actual object. This limits direct manipulation to working with objects within the user's reach. The manipulation itself is done by simply attaching the object to the user's hand when grabbed, which allows the user to translate the object and rotate it based on the movement of the held controller, resulting in a very natural form of interacting with objects. Again this manipulation has its limits concerning the size of the object. When the object is too small or too large, it becomes hard for the user to manipulate it because the object is either too small to easily grab it or so large that it occludes the user's vision during manipulation. If the object is in reach of the user and has a

reasonable size, this technique is very easy to use and needs almost no practice to achieve reasonable results. Due to the user's fatigue and imprecision, however, it lacks precision [11].

3.2 WIM, Worlds In Miniature

The Worlds in Miniature or WIM technique allows the user to interact with objects in an immersive virtual environment by providing a scaled-down version of the user's actual environment (see figure 3.1). This scaled-down environment allows users to select remote objects and interact with them directly. It also enables users to interact with objects which would be occluded from their perspective.

The system created by Stoakly, Conway, and Pausch [19] included a head-mounted display that provides users with an immersive point of view. This head-mounted display was tracked via a *Polhemus* tracker with six degrees of freedom. Additionally the users were given a clipboard that was also tracked with an identical *Polhemus* tracker. This clipboard was placed in the user's non-dominant hand and could be used to control the position and orientation of the WIM representing the floor of a given environment. The dominant hand was outfitted with a tennis ball which had two buttons as well as another *Polhemus* sensor. This device was used to select and manipulate objects.

The graphical representation of the WIM is a scaled-down copy of all the graphical objects in the immersive environment. Whenever the user interacts with an object, whether in the WIM or the actual environment, the corresponding copy would follow. This way the environment itself becomes a widget.

This system allows for a number of application-independent interaction techniques based on the WIM. The additional perspective of the WIM helps to quickly minimize occlusions and selecting objects on the WIM helps users to overcome range as well as occlusion. When manipulating objects, the WIM can allow for a quick, imprecise manipulation but the WIM can potentially also display the user's environment at enlarged scale, then allowing for a more precise manipulation.

3.3 Go-Go Interaction Technique

The Go-Go Interaction Technique mainly addresses the problem of interaction with remote objects. The difference to prior techniques is that it also allows interaction with close objects without changing the technique. This is done by virtually increasing the user's arm length.

Just like the cartoon character which gave this technique its name, it allows users to stretch out their arms in an immersive virtual environment

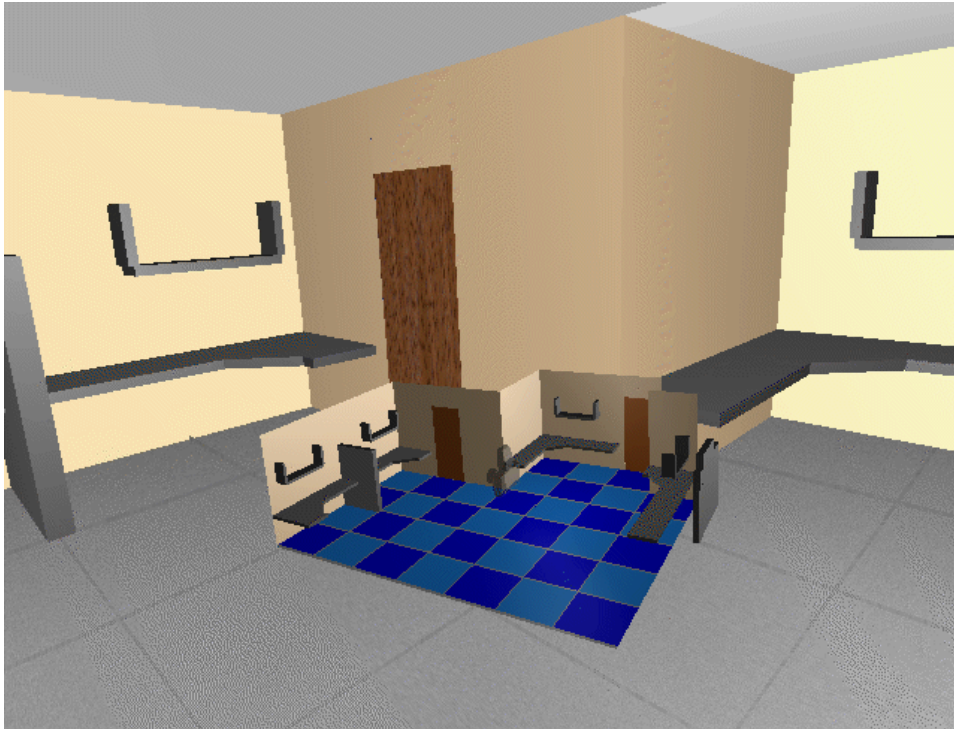


Figure 3.1: A *World in Miniature* viewed against the the represented virtual environment by Stoakley et al. [19].

to reach for distant objects. To render this technique intuitive, seamless, and easy to use, the user's arm length is determined by a non-linear mapping between the user's physical hand and its virtual representation (see figure 3.2).

Poupyrev et al. [17] used two *Polhemus Fastrak* position trackers to track the user's hand in a coordinate system originating at his chest. The position of the virtual hand was then determined by a mapping function which divides the space around the user in two parts. The part closer to the user mapped the hand's movement linearly resulting in the virtual and the real hand having the same position. The part further away from the user was mapped non-linearly allowing the user to move their virtual hand further away than their physical hand. The non-linear space started when the user extended their arm further than two-thirds of its maximum reach.

The evaluation of this technique conducted by Poupyrev et al. [17] has shown that it is highly intuitive. One aspect of its intuitive nature is that it imitates the real-world behaviour of reaching out your hand when trying to grab distant object. Also, due to its modelless approach, it facilitates interaction with distant objects as well as close objects in the same manner.

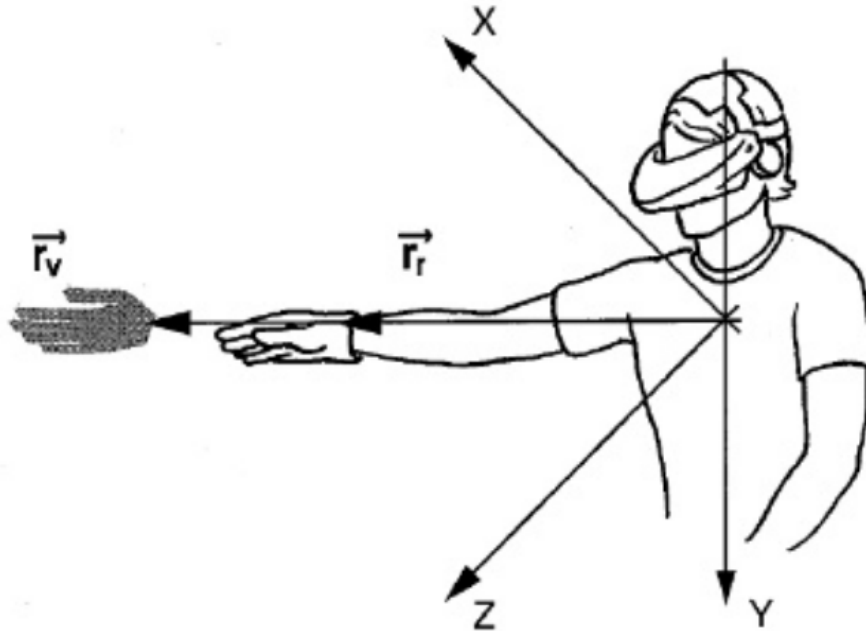


Figure 3.2: Virtual reach r_v compared to the actual reach r_r in the *Go-Go* interaction technique by Poupyrev et al. [17].

3.4 HOMER

The HOMER, Hand-centered Object Manipulation Extending Ray-casting [2] technique is a so called hybrid technique, based on the combination of existing techniques. It has been developed by Doug Bowman and Larry Hodges following a user study using six existing techniques. These techniques were the Go-Go technique, the fast Go-Go technique, the stretch Go-Go technique as well as indirect stretching, ray-casting, and ray-casting with reeling.

The key findings of the study by Bowman et al. [2] are summarized in table 3.1. The study, while slightly favoring the Go-Go technique and the indirect stretching technique, did not establish a clear favorite. However, Bowman and Hodges found that grabbing and manipulation must be considered separately. Based on the results of this user study, the multiple techniques have been developed with the goal of combining features of the prior techniques that worked best. This was done because none of the compared techniques performed well throughout the entire interaction of grabbing, manipulating, and releasing the object.

The techniques that have been previously described use a ray-casting

<u>Technique</u>	<u>Characteristic</u>	<u># subjs.</u>
Go-GO	finite range	7
Go-GO	imprecise grabbing	8
Ray-casting	difficult rotations	11
Ray-casting	can't move objects in/out	11
Ray-casting	ease of grabbing	10
Arm-Extension	ease of manipulation	9

Table 3.1: Number of subjects who commented on selected characteristics of the techniques studied by Bowman et al. [2].

approach for grabbing combined with hand-centered manipulation because these were described as the most effective by almost all subjects. The most basic technique enables users to select or grab an object via a ray-cast and then moving the virtual hand to the position of the selected object as well as attaching it to the hand to allow simple grabbing interaction.

The combination of techniques used by the HOMER technique is based on work by Mine [12] and Wloka [22]. The extensions offered by Bowman and Hodges facilitate quick manipulation of an object on a sphere surrounding the user by determining the grabbed object's position based on the distance of the user's body to their hand, and the current distance of the object while controlling the object's rotation independently. Also the object can be reeled in or out by pressing a button to allow unbound distance. This variation is called the indirect HOMER technique as compared to the prior direct HOMER technique.

Some of the advantages of the HOMER techniques, as described by Bowman and Hodges, are (i) easier and less exhausting object grabbing by using ray-casting instead of the Go-Go technique, (ii) the ability to place objects at any distance, and (iii) more controllable object distance.

3.5 Voodoo Dolls

The Voodoo Dolls technique uses two-handed input to manipulate objects relative to a reference object in an immersive virtual environment. The user chooses the reference object by selecting it with their non-dominant hand. When selecting an object, a copy of this object is then placed in the user's hand. This copy is referred to as the voodoo doll. This voodoo doll is scaled down so that its longest dimension does not exceed 50 centimeters. When moving around this voodoo doll, the original object in the virtual environment does not move as it only serves as a reference to the voodoo doll in the user's dominant hand.

As soon as the user selects an object with their non-dominant hand

the system not only creates the scaled copy in the user's hand, but also creates copies of nearby objects to provide context for the interaction. These additional miniatures are placed relative to the miniature in the user's non-dominant hand (in size and position). With the dominant hand, the user can then select one of the contextual miniatures and move it which then manipulates the original object relatively (see figure 3.3).

In this manner, users are able to always manipulate distant objects the same way independent of their scale. However, the difference in size of the reference object and the manipulated object must not be too great, otherwise it can be hard to manipulate an object because the scaling makes it too small or too large.

The selection of objects in the Voodoo Doll technique, and therefore the creation of dolls, uses an image plane technique. The user's hand has to be placed so that crosshair attached to the hand is placed over an object in their line of sight to select it.

Pierce et al. developed this technique to overcome limitations of previous techniques such as the WIM technique described in section 3.2 or the Go-Go interaction technique described in section 3.3. Their technique is supposed to (i) facilitate users to work at multiple scales using the same way of interaction, (ii) allow manipulation of occluded objects, (iii) take advantage of research showing that a person's dominant hand works in reference to their non-dominant hand, and (iv) simplify working with moving objects [16].

3.6 Proprioception-Based Interaction

The idea of using proprioception in immersive virtual environments is based on the fact that most virtual reality systems lack advanced haptic feedback. Therefore, there is a lack of guidance that would normally be given by the environment. However, one thing users can always feel in virtual environments is their own body. Therefore, proprioception, which describes the sense of position and orientation of a person's body, can be used to function as guidance for interaction in virtual environments.

Mine, Brooks, and Sequin [13] described three forms of interaction in which proprioception can be beneficial for users. When looking at direct manipulation, users can easily determine an object's position and orientation if it is placed in the position their hand due to proprioception. Therefore, it is an easy task to position objects even without viewing the object. Also the placement of objects and controls relative to a user's body can make it easier to find them. This is described by Mine et al. [13] as physical mnemonics and can include interactions such as enabling menu items and placing widgets in the position of a user's hand. Facilitating proprioception also includes gestural actions. For instance, the gesture of throwing an object over a shoulder can be used to delete the object.

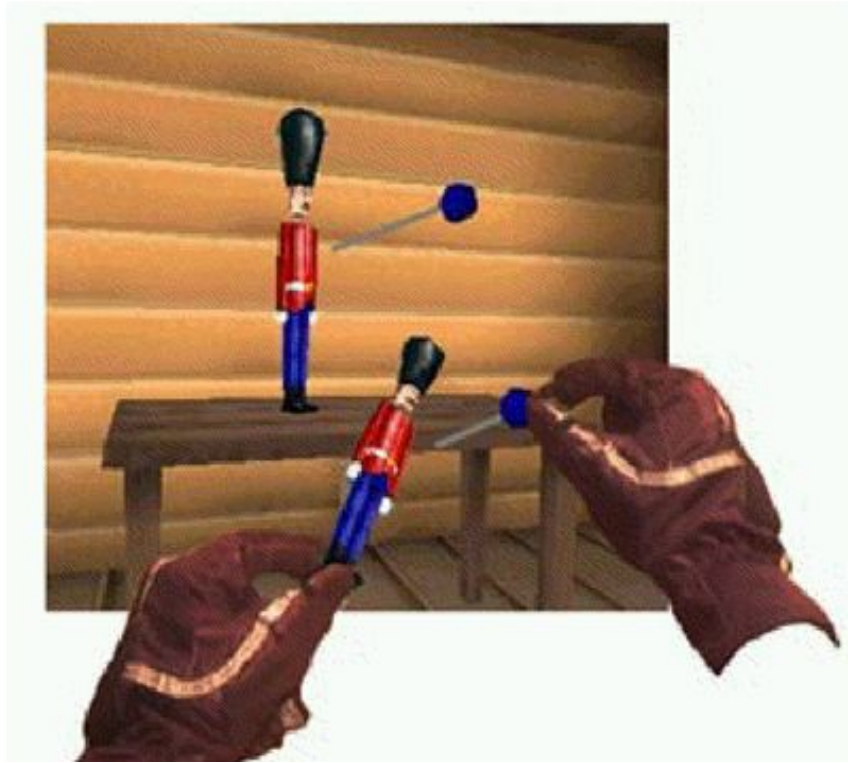


Figure 3.3: Manipulation of a pin and a toy soldier by holding puppets in both hands by Pierce et al. [16].

3.7 Scaled-World Grab

Scaled-World Grab is a technique allowing a user to directly manipulate an object independent of the object's distance to the user. When a user selects an object that is outside of the user's reach the virtual environment around the user is scaled down by a factor determined by the distance of the object to the user. The user can then manipulate the selected object directly in her or his hand. When the object is released, the environment scales back to its former size.

With the object in their hand, users can easily manipulate the object by making use of proprioception described in section 3.6, stereopsis,¹ and head-motion parallax.² This reduces the user's work for manipulating remote objects as there is no need to move towards a remote object to manipulate

¹Stereopsis defines the perception of depth and three-dimensional structure obtained on the basis of visual information.

²A motion parallax driven by head movements.

it, or to move the object to the user's position. Also interaction on close and remote objects works the same, reducing the user's overhead [13].

3.8 SkeweR

The SkeweR technique allows two users to interact with the same object simultaneously in a cooperative virtual environment. There are considerably fewer interaction techniques described for cooperative virtual environments, which makes this technique very interesting.

The skewer technique uses the metaphor of a skewer stuck in the manipulandum³ which is held by two users on either end. The position and orientation of the object is then manipulated based on both users hand position (not the orientation). This technique can also be extended to be used by more users [5].

3.9 PRISM

PRISM, or Precise and Rapid Interaction through Scaled Manipulation, is a technique which is designed to improve a user's precision while directly manipulating objects in an immersive virtual environment. This is done by scaling the hand movement based on the its velocity. The movement can either be scaled up to increase speed or scaled down to improve accuracy. This way the user's intention is interpreted and utilized to seamlessly change the scaling factor. This technique is intended to improve direct manipulation of objects and therefore is suitable to be used in conjunction with other techniques such as WIM (see section 3.2), Go-Go (see section 3.3), HOMER (see section 3.4), or proprioception-based interaction (see section 3.6).

In the implementation of Frees and Kessler [6] a hand-held stylus is used to track the user's hand. The selection of an object is done by intersecting it with the stylus and then pressing a button on the stylus. Then the object can be directly manipulated. During manipulation, a user might want to switch between quickly moving an object to a position and then precisely placing it. To avoid the necessity of menus, key combinations, and other interactions disrupting the current interaction a way of switching between those modes had to be found.

The PRISM interaction technique allows switching between scaled-up and scaled-down movement speed simply by interpreting the velocity of the user's hand. This utilizes the idea that when users try to move an object from one position to another general position they are not necessarily concerned with precision and therefore move their hand quickly. On the other hand, when focusing on precisely placing an object in an exact location users

³An object that is or is to be manipulated.

will normally slow down their movement. This inevitably produces an offset between the object and the user's hand. This offset is recovered in two different ways. Firstly, if the user has accumulated an offset in a certain direction, moving the opposite direction will not move the object until the the hand moved back to the objects position. Secondly, when an offset has been accumulated and the user's hand moves fast enough to trigger scaled up movement, the object slowly catches up with the user's hand by gradually reducing the offset. The scaling of the hand's movement as well as the offset recovery happens per axis, not using the Euclidean velocity. This helps to prevent inadvertent drift.

Object rotation is handled similarly with PRISM. Very subtle and slow rotations are ignored. When the rotation speed increases, the object is first rotated proportional to the distance the hand is rotated. An even faster rotation triggers direct rotation of the object with the hand rotation followed by offset recovery on even faster rotation. Quaternions are used to calculate the rotation; this helps avoiding problems with Euler angles like the Gimble lock. Frees et al. have also suggested their approach of scaled rotational speed can also be used to improve precision when using ray-casting for object selection [7].

3.10 3-Point++

The goal of the 3-Point++ technique for direct manipulation is to provide a way of precisely manipulating large objects in immersive virtual environments, which can be problematic because of the objects occluding the users vision. The 3-Point++ technique uses a tool that consists of four points attached to the manipulandum. Three of these points are placed on a plane and are not aligned, the fourth point is the barycenter of those.

The idea behind this technique is that the precise position and orientation of an object are described by those three handle points. By moving those three points the new position of the object can be calculated. If one handle point is moved by the user, the object can rotate around the axis created by the remaining two points (see figure 3.4). If two handle points are moved, the object can rotate freely around that point. Based on the new position of the handle points, the barycenter, and the new position of the object, are constantly being calculated. The object can be manipulated approximately by using the barycenter handle, which allows direct manipulation of the object by attaching it to the user's hand. The handle points can also be locked and unlocked giving the user the option to constrain movement of the remaining points.

An evaluation has been conducted by Nguyen and Duval which compared direct manipulation as seen in section 3.1 with their 3-Point++ technique. While comparing speed and accuracy of the users, the result of this study

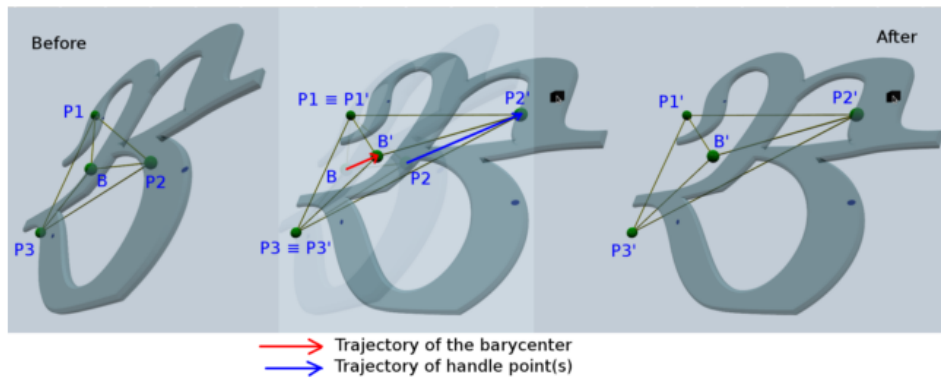


Figure 3.4: Example of the 3-Point++ technique [14]. The handle point P2 is manipulated, the object is rotated around an axis created by the two handle points P1 and P3.

shows that the 3-Point++ technique has not improved average manipulation time. They suggested this might be due to the complexity of the interaction using the 3-Point++ tool [14].

3.11 Handle Bar Metaphor

The Handle Bar metaphor was developed for use with a traditional screen and a depth sensor (e.g., *Kinect*⁴) to allow mid-air interaction with three-dimensional objects on the screen without using an input device. It uses the position of both of the user's hands to translate and rotate the virtual object. This is done with the metaphor of a handlebar that is pierced through the manipulandum. To manipulate the object, the user now manipulates the virtual handle bar and the object follows. Because of the missing rotational information of the hands in this setup, the handle bar cannot be rotated along its own axis (roll). The rotation around the remaining axis is possible, however, by just using the positional information. Also by increasing or decreasing the distance between the hands, the object can be scaled [18]. Despite the metaphor having been developed for a different setup than an immersive virtual environment, it could also improve precision when manipulating objects in such an environment.

⁴A motion sensing input device.

Chapter 4

Interaction Design

Based on the knowledge gained in the previous chapters 2 and 3, an interaction design for improved precision during object manipulation will be proposed. After examining suggestions to improve object manipulation in immersive virtual environments, as well as general peculiarities of interaction design in immersive virtual environments, the approach chosen for the design of PROMT will be explained.

The goal of this interaction technique is to enable users to manipulate three-dimensional objects in an immersive virtual environment. The difference in available input devices is an important reason for the substantial difference of interaction in virtual reality compared to interaction in a desktop environment. Stereoscopic vision instead of two-dimensional projections on a display further increases this difference. While interactions in virtual environments enhance intuitive interactions, they do not offer precision comparable to traditional desktop interaction.

Object manipulation in this context mainly describes the manipulation of three features of an object: translation, which defines the position of an object in a three-dimensional space; the rotation of an object in a three-dimensional space around a certain point; and an objects scale that defines the size of an object in relation to its surroundings. Additional to these manipulations, a basic manipulation tool may have other necessary features such as the selection of an object and the movement of the viewpoint.

4.1 Interaction Design for Immersive Virtual Environments

Interaction design for virtual reality applications is a very interesting topic at the moment. The reason for this is mainly because developers of virtual reality applications are just now finding out which interaction techniques are feasible and which are not. Best practice approaches for user interaction

in a desktop environment have already been explored and tested over years of actual use. Techniques that work well have been established through their survival in an open market. In immersive virtual environments, the actual market has been comparably small since their first appearance.

Only very recently has the market started to grow rapidly. This means that user interaction paradigms are now going to be put to the test and therefore will start to experience quick adaptations based on how successful they are. So far even very basic interactions which are needed for a large number of applications do not yet have a set interaction technique and while virtual reality games are already becoming very popular, the number of professional applications is still rather low. Now is the time to find best practice approaches for a lot of different interactions in immersive virtual environments.

4.1.1 Stereoscopic vision

One big advantage of immersive virtual environments is that users are able to experience three-dimensional environments in a stereoscopic fashion rather than projected on a two-dimensional screen. This improves not only the immersion but also the perception of three-dimensional objects, thus improving the accuracy of a user's image of a virtual object.

4.1.2 Intuitiveness vs Precision

Controller based interaction in immersive virtual environments has improved intuitive interaction with three-dimensional objects. The ability to manipulate the input device in three dimensions and six degrees of freedom brings a user closer to an interaction similar to his everyday interaction with physical objects. It feels more natural to position a three-dimensionally tracked controller close to another object in the same space, rather than to use a two-dimensionally tracked mouse. However, this does not improve precision. In regards to this issue, techniques to precisely move objects in virtual environments while maintaining speed and ease of use have not been established. An experiment conducted by McMahan et al.[10] suggests that three-dimensional input devices improve the performance of manipulation tasks making it plausible that combined with improved precision virtual environments could improve overall performance in industrial applications including three-dimensional object manipulation tasks.

4.1.3 Menu Interaction

Most desktop applications have a need for some form of menu. The way users interact with those menus is very well known and tested. However, in virtual reality applications, traditional menus are not applicable. The menu has to be implemented in the three-dimensional space of the virtual environment.

Also forms of heads-up displays are hard to use because they either distract the user's vision if they are close to the middle of the user's field of view, or they are hard to view because they are at the edge of the user's field of view.

4.2 Use Case

After interviewing students of the University of Arts and Industrial Design in Linz, a use case was created. The use case is an application aimed at the conceptual phase of industrial design. It should enable designers to quickly create three-dimensional models made up of primitives. This should enable users to quickly block in the basic shape of a given design and evaluate its feasibility in an immersive virtual environment.

This approach of blocking in is taken from a common design work flow, in which the first step of a design is to block in a basic shape of the design at the beginning, to quickly evaluate multiple possible basic designs. This is often done with materials that can quickly be formed and are cheap, so discarding designs is not a problem. Therefore implementing this step in a virtual reality application can be beneficial for reducing material usage in the early stage of designing an object.

Furthermore, the application focuses on objects larger than regular screen size to improve the user's perception of objects by displaying them in live size rather than scaling them down on a screen. Also the objects should not be too large. When viewing objects larger than room size, it becomes hard to perceive proportions.

The application should therefore allow the user to spawn multiple three-dimensional primitives via a three-dimensional user interface. Also it should enable the user to select them, and freely manipulate them. While manipulating an object, the goal should be to quickly move the object to its approximate scale, position and orientation, to allow the user to adjust those features precisely afterwards. The application should also feature a widget to display an object's position and orientation via visualized local axis, and numerical representation of the objects transformation. To enhance precision a way of separating the degrees of freedom could be implemented as well as further precision enhancing techniques.

4.3 Included Features

The core features needed for PROMT, as an interaction technique for object manipulation, focus on the translation, rotation, and scaling of object as well as the separation of degrees of freedom. An interaction technique for object manipulation however, is also closely coupled with other features of an application surrounding the manipulation. Therefore, other interactions

such as the selection of objects and menu interaction will be considered. Those features are defined by the need of the use case described earlier in section 4.2. The additional features are mainly based on existing techniques.

4.3.1 Manipulation

The central feature of PROMT is the manipulation of objects. Considering the use case 4.2, the manipulation has to be optimized for room scale interaction. This means users must be able to manipulate objects within their reach as well as objects further than an arms length away. To accomplish this, either two separate interaction approaches have to be usable simultaneously or an interaction approach which is usable in close proximity as well as for remote interaction has to be found.

To overcome the problem of precise manipulation of three-dimensional objects in a mouse-based manipulation, Houde [9] suggested a widget-based approach called the handle box. This approach utilizes the separation of degrees of freedom, or DOF.

DOF Separation

A common solution for mouse-based object manipulation in three-dimensional space includes the separation of the degrees of freedom by manipulating the object on an isolated axis or plane. This is important in mouse-based object manipulation especially, as a two-dimensional input has to be mapped to manipulate objects in three-dimensional space. The handle box mentioned above is a box surrounding a three-dimensional virtual object with handles attached to it. They are used to constrain geometric transformation with the handles representing axes or planes in which the object can be manipulated.

Beside helping the user to overcome the problem of a two-dimensional input which is mapped to manipulate objects in three-dimensional space, this also improves precise manipulation of the object by preventing the user to accidentally manipulate the object in an unintended way.

Object manipulation in a controller-based immersive virtual environment allows a user to directly manipulate an object's translation and orientation based on the movement and rotation of the controller. This eliminates the need for axis separation. However, by adding axis separation the precision during object manipulation can be increased. An evaluation conducted by Mendes et al. [11] compares a direct six degrees of freedom manipulation described in section 3.1, the PRISM technique described in section 3.9, and a widget-based interaction approach separating the degrees of freedom. The evaluation suggests that the separation of the degrees of freedom can improve precision compared to the other tested techniques. This comes at the cost of a higher task completion time for complex tasks. For applications which rely on precise manipulations however, this increased task completion time

can be tolerated.

Translation

When viewing direct manipulation (see section 3.1) the translation of an object in immersive virtual environments is very intuitive in controller-based interaction. The virtual object is simply attached to the user's controller and moves just as the controller does. This interaction has proven to be very intuitive. Users very quickly understand the concept and have almost no problems using it because it resembles natural interaction with object very closely. This approach however limits the precision of the translation. Even with very little error in the tracking system the, user's own imprecision cannot be eliminated using this technique. In addition a user's fatigue during manipulation has to be taken into account. When users have to hold up their arms for longer periods of time while also holding controllers in their hands, the precision decreases with increasing fatigue.

Therefore, means of improving precision during translation tasks have to be found. The PRISM interaction technique (see section 3.9) suggests scaling down the movement of the virtual hand when the user moves their hand slowly. This technique has shown to be well accepted by users and improve precision in translation tasks without sacrificing speed as suggested by guidelines for mid-air object manipulation by Mendes et al. [11]. Also widget-based approaches have been suggested like the 3-Point++ technique (see section 3.10) which was less well accepted by users because of its complexity.

Other techniques try to overcome the problem of translating objects which are further than an arm's reach away from the user. Direct manipulation (see section 3.1) is limited in range because the object cannot be translated further than the user can move his hand. The Go-Go interaction technique (see section 3.3) uses a virtually extended arm to improve the user's reach for selecting and manipulating distant objects. Also techniques based on ray-casting have been developed, allowing users to translate objects as if attached to rod.

Rotation

Similarly to the position of objects, their orientation can also be manipulated using direct manipulation (see section 3.1). This behaves just as naturally as translation does by simply rotating the object in an immersive virtual environment by the same amount the user's actual hand rotates. Naturally this is limited by the user's inability to rotate their hand indefinitely, especially when performing long rotational operations. The precision of performed rotations is also limited in precision, again due to the user's imprecision and fatigue.

An input device for immersive virtual environments that allows rotational input is advantageous for rotational operations. Techniques using input devices that only track the position of a user's hand, rather than position as well as orientation, mostly rely on two handed interaction. One example for such a technique is the 6D Hands technique suggested by Wang et al. [21]. Directly manipulating the rotation by rotating the hand however has found to be practical only for rough transformations as a study conducted by Mendes et al. [11] suggests.

Scaling

Scaling, compared to translation and rotation, lacks a direct real world counterpart as quickly changing an object's size is not a natural operation. However, in modeling and design applications, scaling is a very common manipulation task. This is also the reason why Bowman et al. included it as a basic interaction task. [3] The operation of scaling an object not only lacks a natural counterpart, it also expresses a degree of freedom which cannot be represented simply by the position and orientation of a tracked controller in three-dimensional space.

To represent scaling, at least some kind of additional interaction or gesture is needed. In a controller-based environment this gesture will often include two-handed interaction. For instance, the metaphor of holding an object in both hands and pulling it apart could represent the intent of scaling the object up.

Techniques like the handle-bar technique [18] utilize this metaphor for scaling as described before. A widget-based approach including scaling was suggested by Conner et al. [4] This approach includes three-dimensional handles attached to the object. They can be directly manipulated intuitively. However, each handle stands for an axis which, when selected, is the only axis in which the operation will take place. This separation of degrees of freedom, as described earlier in section 4.3.1, allows for not only uniform scaling but scaling in each axis of the object.

Discrete Manipulation

A way of helping users manipulate objects precisely is to limit object positions to discrete steps. By only allowing movement in discrete steps, the user loses the freedom of placing an object in any arbitrary position. However, they gain speed and precision. This can be helpful when the alignment to another object is more important than the exact distance between the objects. Also discrete steps in the orientation can be used to quickly place multiple objects at the same angle.

In desktop environments, three-dimensional modeling applications often utilize a form of snapping [1] which either places an object at points of a

defined snapping-grid or aligns them according to already existing objects. Snapping systems using a grid as constraint are commercially successful, due to the fact that they are easy to learn and implement [1, p. 239]. In mouse-based systems, snapping systems are limited to a two-dimensional plane due to the capabilities of a mouse. To work around this restriction, snapping in mouse-based systems is strongly connected to the camera position in the three-dimensional scene. Often orthogonal views are used to clarify in which plane the snapping should occur.

In immersive virtual environments, positional input is possible in three dimensions, eliminating the need to use the camera position to precisely determine where snapping should occur. Therefore snapping, especially restricted to a three-dimensional grid, can be used more naturally. This helps to prevent interruptions of the workflow during manipulation.

4.3.2 Selection

The selection of objects or menu items is a well defined task in mouse-based interfaces. Usually a mouse pointer is used to select items by hovering over them and clicking. This applies for two-dimensional objects, such as a menu item or a two-dimensional graphic. In applications dealing with three-dimensional objects and scenes, this is done by determining the position of objects in screen space. Problems like occlusion can easily occur when combining two-dimensional and three-dimensional space, as the users view and interaction is always limited to a plane.

A three-dimensional input device could potentially nullify those problems, as interaction with objects happens in the same space as the presentation of these objects. The user's vision, however, still suffers from occlusion in immersive virtual environments. Also certain techniques like selection via ray-casting, or other vector-based pointing techniques, can have problems with occlusions. To overcome the problem of selecting an occluded object, techniques like the Go-Go technique (see section 3.3) or the simple virtual hand technique (see section 3.1) enable users to select objects by overlapping their virtual hand with the object they want to select.

Another topic of selection techniques is the problem of selecting multiple objects at the same time. A common solution in mouse-based interfaces is to click and drag a rectangle over all the objects to select. The immersive virtual environment equivalent to this area-based selection is volume-based selection. One technique that allows volume-based selection is the flashlight technique [3, p. 285]. It enables users to cast a cone from their input device, similar to the cone of light cast by a flashlight. Objects within this cone are then selected. An advanced technique similar to this is the aperture technique [3, p. 286]. Just like the flashlight technique this technique casts a cone from the input device, additionally the user is able to adjust a virtual aperture to control the angle of the cone.

Volume-based selection can also be achieved via the sphere-casting technique [3, p. 287]. This technique enables users to cast a predefined volume into the virtual environment, which then allows the selection of all items in this volume. The position of this volume is determined by the intersection point of a ray-cast with an object in the virtual environment.

4.3.3 System Controls

The interaction within an application is rarely limited to its main task. To allow sophisticated interaction in an application, users do not only need a method of interaction for the main task, but also to control the system around this task. In a drawing application, users may not only want to draw on an empty canvas; they also want to select a certain brush, adjust its parameters, add layers, save the work they created, and name it. To achieve this, system controls are necessary.

In desktop environments, the problem of system controls seems trivial, as it is already well developed. Two-dimensional user interfaces enable users to use the mouse to activate menu items, drag and drop objects and files, and select objects. Additionally the keyboard can be used for alphanumeric input and offers enough buttons for shortcuts. Even though many techniques for three-dimensional system controls exist, it can still be challenging to implement simple tasks that are well known in desktop environments.

While many techniques from two-dimensional interfaces have been adapted in three-dimensional interfaces, which is suitable for three-dimensional interfaces in desktop environments, they may not always be effective in immersive virtual environments. The-six-degrees-of-freedom input in immersive virtual environments may lead to some problems when compared to traditional input but also creates new possibilities.

Because most users are already using graphical interfaces in desktop environments, it has become common for developers to use them in immersive virtual environments as well. This may happen by rendering a plane containing a very traditional two-dimensional graphical interface and using ray-casting as the equivalent of a mouse pointer. Other attempts try to use graphical interfaces which are more adapted to the virtual environment, such as interfaces with one degree of freedom or three-dimensional widgets.

4.4 Design of PROMT

In the last sections of this chapter, general problems and advantages concerning interaction design in immersive virtual environments have been described. Furthermore, the use case in which PROMT will be embedded and tested has been specified, to then create a list of features and necessities for the use case as well as the technique itself. As the main interaction of this use case is the manipulation of three-dimensional primitives, the main focus

is set on interactions regarding this task. The interaction's design beside the main task will be based mainly on existing interaction designs, such as the selection and system control features.

In this section, the design decisions for PROMT and its use case will be explained, starting with the manipulation technique itself. This explanation will include encountered problems during early stages of development and the changes they led to.

4.4.1 Overview

For PROMT, an approach that is comprised of multiple different kinds of interactions has been chosen. It is based on parts of techniques that have proven to be intuitive and practical. The technique is based around a widget which is placed on the position of the user's controller as well as a context-sensitive widget placed on the selected object. It is activated as soon as the user selects an object via a ray-cast. The widget then displays the axis of the object's local coordinate system. By highlighting one of those, the user gets feedback about which one is selected and therefore can be manipulated. Selection of an axis occurs via a gestural input. The user aligns the controller's orientation with the orientation of one of the axis to select it. Also the shape of the axis representation gives feedback regarding the manipulation mode the user is in. This can be rotate, translate, or scale. The user then aligns his or her controller with one of the selectable axis and locks in the axis as selected by holding down the grip buttons on the controller. Based on the chosen manipulation, a gesture like sliding the controller along an axis or rotating it via a crank will start the manipulation. A step size can be chosen to limit positioning and rotation to a preferred discrete size.

4.4.2 Input device

As described in section 2.2, the input device plays a significant role in interaction design. It defines how users ultimately interact with an interface. So to achieve precise interaction in PROMT, an input device that enables precise input in terms of positional tracking and functional input, as well as a reliable tracking system, was chosen. These prerequisites favor a controller-based system like the one used in the *HTC Vive* system, which was chosen over systems like the *Leap Motion* system. Figure 4.1 shows a schematic representation of the vive controllers showing its button layout as well as its tracking sensors.

4.4.3 Widget

A widget-based manipulation approach has been chosen because of its potential in increasing precision while decreasing the amount of unwanted actions. In this application, a context-sensitive widget is used. The widget is placed

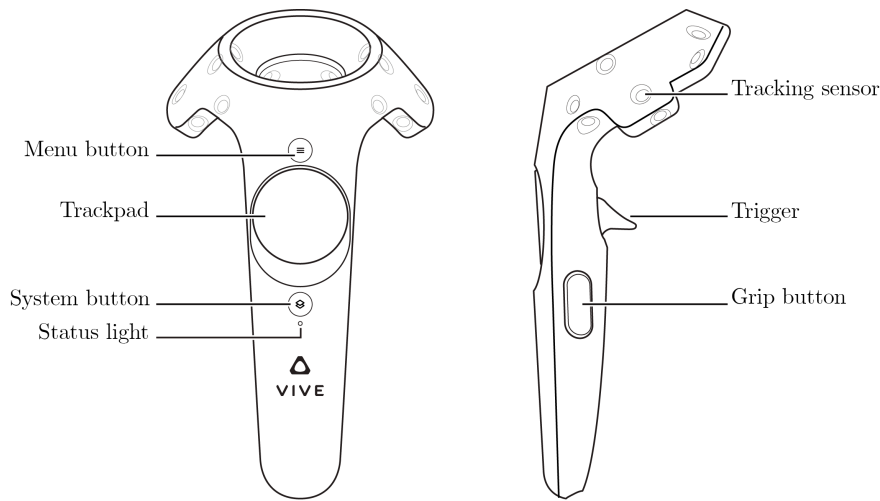


Figure 4.1: Button layout of the *HTC Vive* controller.

directly on the object that is manipulated. This object forms the widget's context and therefore the widget is adapted to the object in terms of size and orientation, as well as its position. The widget is used as a visual feedback for the user. It allows for improved judgment over the object's position and orientation. The visual feedback also provides information about if a certain axis is selected separately, and if so the selected axis is highlighted as well. Furthermore, it provides feedback to the user about what manipulation mode the tool is set to, as seen in figure 4.2. This approach was chosen for multiple reasons. Firstly, in traditional (non-immersive) three-dimensional modeling applications, this form of widget-based approach is well known and has proven practical. This allows experienced users to quickly learn to use this adapted approach. Secondly, it allows a precise and understandable way of axis separation. Because a simple adaptation is not possible due to the difference in input devices, different forms of interaction were added to improve the usability of the widget-based approach in immersive virtual environments.

4.4.4 Axis Alignment

One of the most important features of PROMT is the separation of degrees of freedom. This approach has been proven to increase precision during object manipulation. The way in which it was implemented in existing techniques was however not fitting for PROMT. Therefore a gestural interaction for a controller-based environment was chosen. The gestural interaction focuses on aligning the used controller with the axis the user wants to select as

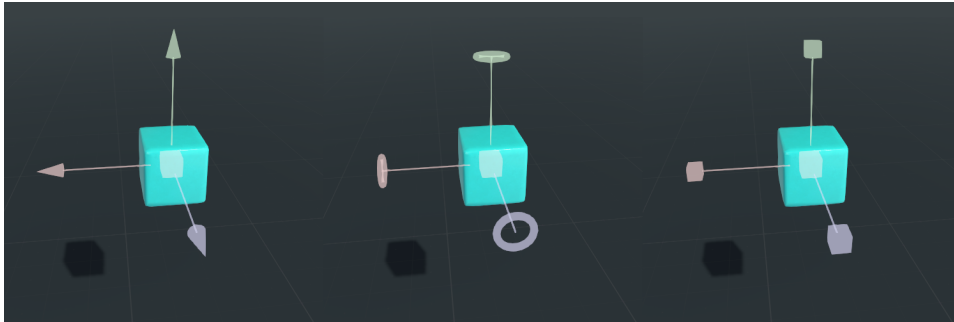


Figure 4.2: The widget created for PROMT during translation, rotation, and scaling mode.

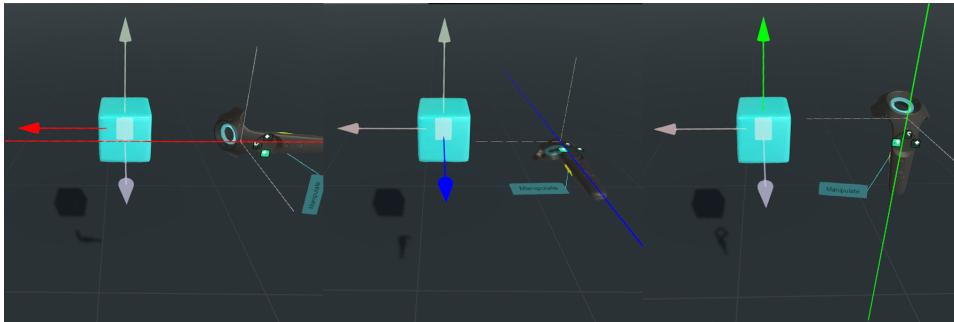


Figure 4.3: Selection of all three axis separately via axis alignment.

seen in figure 4.3. This allows users to quickly switch between manipulating separate axis without breaking their flow and without forcing them to shift their focus. To aid this idea visually, a widget is placed in the user's hand which represents the orientation of the selected object. This helps the user to align his or her controller correctly. Additionally the user can also select the axis also by touching them directly with the controller. This option is not suggested, but it has proven that users prefer to have this option available as well.

Directly manipulating objects by attaching it to a user's hand has proven so intuitive and practical that even though it is not the main way of interaction in PROMT it had to be added for the use case. Placing relatively small objects quickly without focus on precision can be done using direct manipulation. This way users can also change position and orientation at the same time.

4.4.5 Translation

PROMT is mainly based on separating the mode of manipulation as well as the axis in which manipulation takes place. For translations, a metaphor of the user grabbing a rod was developed. This rod represented the axis, along which the user pulls or pushes the object. This allows for a quick flow from aligning the controller with the axis confirming the selection and directly starting translation. Additionally parts of the PRISM (see section 3.9) interaction technique developed by Frees et al. [6] were adapted to scale the amount of translation based on the speed of the user's hand. This improves precision when moving the hand slowly, as well as speed when moving the hand quickly.

4.4.6 Rotation

The metaphor for rotating an object along an axis has changed multiple times due to problems in early user tests. The first solution utilized a touch pad placed on the controller. The touch pad was used to recognize a rotational motion of the user's thumb on the pad. Based on this motion, the object would rotate around the selected axis. This approach was not used because users found it to be not intuitive and also it put excessive strain on the thumb joint when used even for a short time. Another approach was closely related to the metaphor of holding a rod used in translation. In this approach, the rod had to be turned instead of pushed which was found to be intuitive by most users. Unfortunately tests have shown that this approach puts too much strain on the user's wrist because the hand is often in a non perfect position to be rotated after aligning it with an axis.

Ultimately the crank metaphor was created to overcome these problems. This metaphor lets the user rotate the object as if it were attached to a crank. By using this metaphor it was possible to minimize the strain on users' joints during interaction while maintaining an intuitive interaction. By increasing the distance between the user's hand and the starting point of the crank, the precision of the user's input can also be increased intuitively. To help users understand this metaphor, an additional widget has been introduced. As seen in figure 4.4, it consists of an axis, an indicator, and two circles. The smaller circle is marking the minimal distance a user has to have to the center of the widget before interaction starts. This threshold stops the user from unintentionally beginning a rotation before its intended. The bigger circle shows the distance of the user's hand to the center of the widget.

4.4.7 Scale

To scale an object, an approach very similar to the translation metaphor was taken. Initially an approach requiring two hands to use was considered. Such approaches for scaling have been used before in other interaction tech-

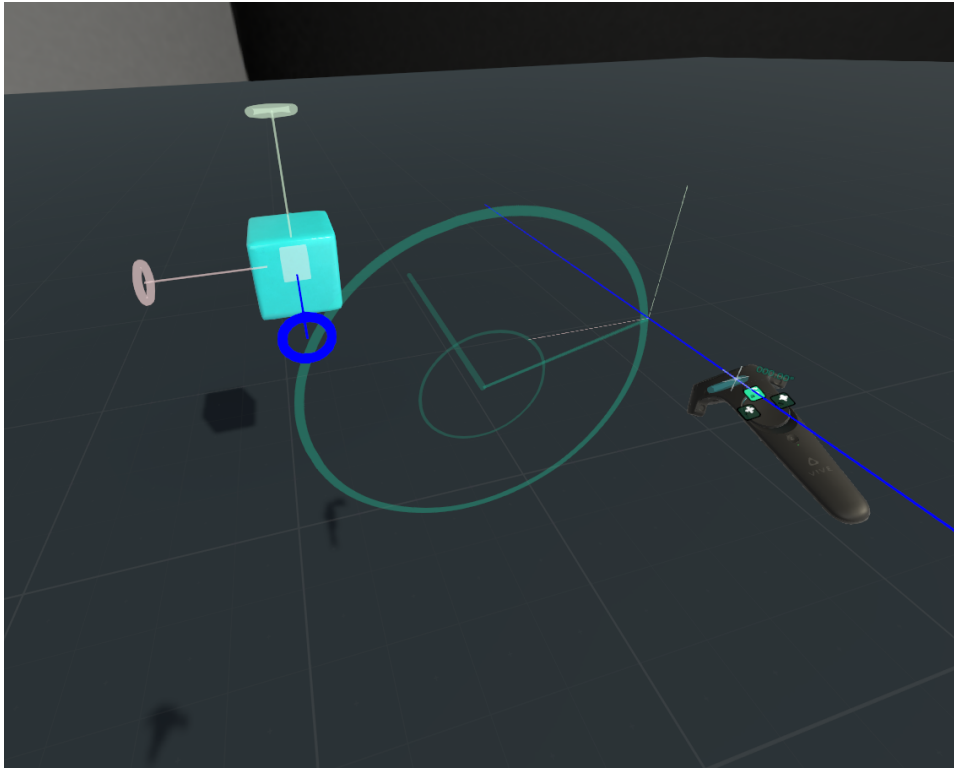


Figure 4.4: Screenshot depicting the crank handle widget and interaction.

niques. However, because it was decided to focus PROMT on one-handed interaction, to allow the second hand to be used for other interaction, a different approach was preferred. Due to the similarity in interaction, it seemed more practical to let the user scale objects by also using the metaphor of pushing and pulling a rod attached to the object. This approach also felt very intuitive and was accepted well in early user tests.

4.4.8 Snapping Grid

Due to the fact that users have problems keeping their hands as still as possible while interacting in mid-air, and also because the tracking of the controllers have small deviations, a way of minimizing those factors was needed. Using a filtering process seemed redundant because the position was already being filtered by the tracking system used. Therefore another approach used in certain design applications similar to the snapping dragging approach [1] was selected. This approach was to give users the option to activate snapping in a three-dimensional grid. This has proven so effective that it was decided to always activate this discrete manipulation and just

set the base step size small enough so it was not apparent to the user. This improved the ability to hold an object still in mid air. Also it presented the possibility to give haptic feedback to the user whenever the object moves to a new step. This was done to give the user a better way of assessing how far the object was moved.

Snapping to a global grid in three-dimensional space combined with axis-wise manipulation has proven problematic in certain situations. When freely translating the object in all three axis, the snapping works as intended; when the object is closer to a new snapping point than to the last one, it will change its position to the new snapping point. This happens for each axis separately. When the object is moved along a particular axis not in line with the global snapping grid, the object will not move along this axis but jump between the grid points located around this axis. This movement is very irritating and prevents the user from clearly seeing where the object is moving.

To prevent this jumping motion of the object, the snapping feature was separated into two parts. When moving the object in all axis simultaneously, the object snaps to a global grid which is the same grid for every object; when only a single axis is manipulated, the object moves strictly along this single axis by one step size at a time. This means the object may lose alignment with the global grid when moved along a single axis; in return the object moves only along the selected axis in a straight line. This was necessary to prevent the object from jumping in all directions when trying to move it along a single axis.

4.4.9 Selection

The selection of objects is not a part of the main interaction but directly connected to it. Therefore it had to be simple enough so it would not challenge the user additionally between object manipulation tasks. One technique for object selection that was easy to learn and understand is the ray-casting technique. It allows for simple, single-object selection by using the hand-held controller similar to an actual laser pointer. This selection technique is effective when selecting objects out of the user's reach. On the other hand, it can be impractical when used to select objects close to the user. Ray casting can also be impractical when selecting small objects at a long distance as it forces the user to be very accurate as seen in figure 4.5.

To prevent the user from having to use ray-casting to select close objects, a secondary way of selection can be practical. Therefore the possibility to touch the object to select it was added. This allows users to quickly select objects close to them in an intuitive fashion.

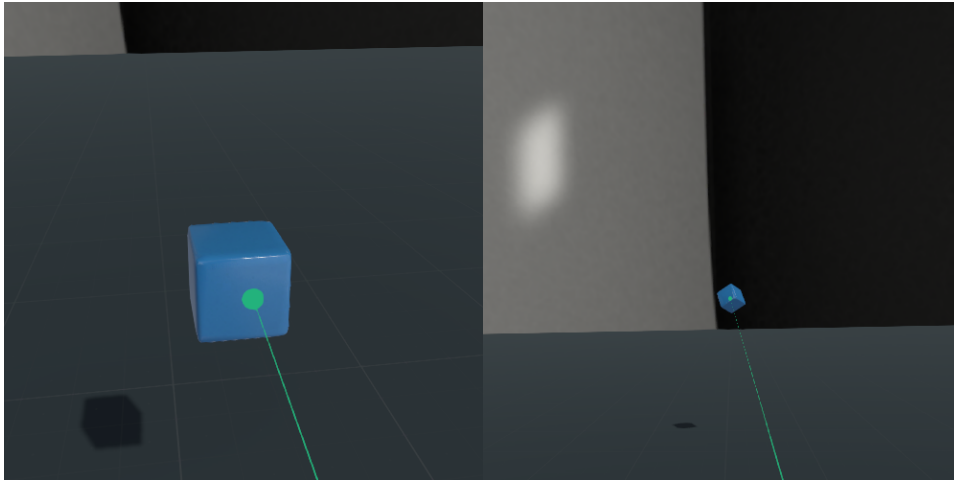


Figure 4.5: Screenshot showing the selection of an object via ray casting at medium and long distance.

4.4.10 System Controls

The placement of system controls in immersive virtual environments is not trivial, as explained in section 4.3.3. The use case that was created only has a relatively small set of system functionality, still the interaction with system controls has proven difficult. To accommodate all the controls needed, a multi modal technique was chosen as described by Doug Bowman et al. [3, p. 385]. A multi modal technique in terms of system controls is described as a technique comprised of multiple basic types of techniques.

The context-sensitive widget used in PROMT is already a form of system control. It allows users to select an object's axis by touching the widget directly with the controller. Additionally the axis alignment is a gestural technique that allows the user to select separate axis without the need of being close to the object.

When possible, system controls were placed on hardware buttons of the controllers. However, due to the limited number of available buttons per controller and the fact that they are not designed for specific interaction techniques but broad use, they only offered very little possibility for assignment of system controls. So it was decided to put the system controls that are likely to be used the most on the available buttons. To select which manipulation mode to use, users can utilize the touch-pad on top of the controller. The modes are selected by pressing the left, top, or right side of the touch-pad. To clarify to the user which mode is active and where to press to change modes, the positions are indicated in the immersive virtual environment by icons. The second system control that was placed on a

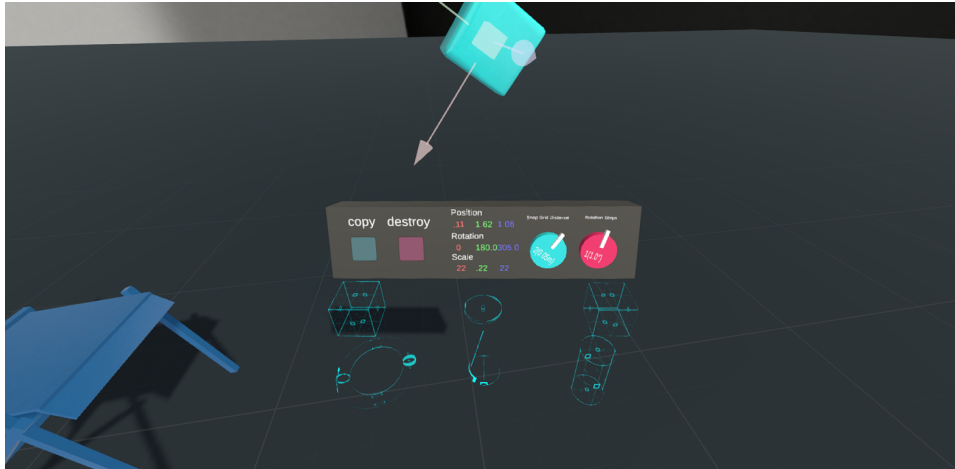


Figure 4.6: Screenshot showing the three-dimensional graphical user interface.

hardware button is the menu toggle. This toggle lets the user show or hide a three-dimensional graphical user interface. This menu is placed relatively to the user's head position to appear at a comfortable height to be reached with the user's right hand.

The three-dimensional graphical user interface, as seen in figure 4.6, is the third type of system controls used in this application. On the top of this graphical user interface, a form of adapted two-dimensional user interface is used to call system functions such as copy and delete by pressing two-dimensional buttons using a visualized ray-cast as pointer. Next to this area are two dials used to change the size of rotational and translational steps during manipulation. These dials can be seen as a one degree of freedom menu. Underneath this part of the menu prototypical primitives are displayed similar to the way they would be displayed in a shelf. The user can simply drag them from the menu to spawn a new primitive in the scene.

4.4.11 Scaled Input

The varying distance at which users can manipulate objects using PROMT contributed to the problem of translation speed. When objects are very close, moving them at the same speed as the actual controller is acceptable; however when moving objects at greater distance, the direct translation of controller movement to object movement can become tedious. Similarly to the way this was approached in the PRISM technique, described in section 3.9, a scaled input was added to translational interactions. By monitoring the current velocity of the controller, the application decides on a factor to scale the movement of the controller up before applying it to the object. This en-

ables the user to quickly move objects in an imprecise way, when precision is not intended.

Just like scaling up the user's movements to increase speed, scaling down his movements can be used to increase precision. When the velocity of the controller drops under a certain value, the velocity of the manipulated object is scaled down. This makes up for jitter in the tracking system as well as human fatigue.

The same could be done for rotational operations. However, according to Frees et al. [7] scaled input is not appealing for rotation as it has a confusing effect on users. Therefore using scaled input for rotational operations was not implemented.

After deciding on the interaction design of PROMT and its use case, the following chapter will describe the implementation of the actual application. The final implementation is based on the testing applications used to find the final design.

Chapter 5

Implementation

The last chapter reviewed the interaction design of PROMT and its use case application. During this review, the parts of the interaction design that were changed due to different flaws in their design have been described. After reaching the final design, this chapter will now describe the technical implementation and hardware setup. It will also discuss which hardware and which development environment was chosen and why. Furthermore, it will include an overview of the application architecture and a detailed description of how it functions.

5.1 Technical Setup

5.1.1 Development Environment

When choosing the development environment, multiple factors relevant to the goal were considered based on the set requirements. This goal was to create a prototypical application, used as a proof of concept for the interaction technique that has been developed. The main focus when deciding was to use an environment that includes basic features to create a three-dimensional application. Also virtual reality headset support was an important factor, as well as the familiarity of the environment.

Based on those factors, and the fact that experience was gained in a previous virtual reality project, the *Unity*¹ engine was chosen. It allows for quick prototyping and includes packages that implement character controllers for virtual reality applications. It also supports current consumer grade headsets. This allowed for a shortened development time and focusing on the parts relevant to the interaction design of PROMT.

¹<https://unity3d.com>, last visited 13.09.2017



Figure 5.1: Representation of a typical *HTC Vive* setup.

5.1.2 Virtual Reality Setup

As input and output devices, the application required a virtual reality system including a virtual reality headset and controllers that are tracked in six degrees of freedom. The controllers form the main input device of this interaction technique. Also a room-scale tracking was beneficial as the use case aimed at manipulating object up to room scale. As those features are supported by the current *HTC Vive* virtual reality system, and because of its availability, it was used to develop PROMT.

The *HTC Vive* system includes a virtual reality headset, two hand-held controllers, and two *Lighthouses* used for tracking. The system, as seen in figure 5.1, uses a *inside-out tracking* technology. Photoelectronic sensors are placed on the tracked objects. These sensors receive infra-red light emitted from the *Lighthouses*, which are placed in two opposing corners of the tracked room. The headset is connected to a computer via cables; the controllers are wireless and connect to the headset via *Bluetooth*.

5.2 Application Architecture

As seen in figure 5.2 the application containing the use case, as well as PROMT, is consisting of three main parts. The first part is the *Steam VR* plugin, which is a collection of tools and services for virtual reality applications based on the *Open VR* library. This is used to communicate with

the head-mounted display and controllers. The second part is the *Virtual Reality Toolkit* or *VRTK*. The *VRTK* contains a collection of tools to allow the use of basic interaction techniques for immersive virtual environments. These tools were not used in this project; it utilizes the layer of abstraction the *VRTK* provides to develop the application without direct dependencies to a certain hardware.

PROMT is closely connected to the *VRTK* to allow hardware independent input. The selection manager, which is a Singleton class managing the selection of objects and axis, as well as the manipulation manager, which is keeping track of all manipulations applied to an object, are separated from the rest of the system. The other parts are components attached to the *VRTK* controller script to follow the logical and hierarchical structure of the *VRTK* system. The controller script entity sends the input given by the controller to a tool manager. The tool manager uses the state pattern to then interpret the input depending on the selected tool. This structure potentially allows the addition of other tools. For this project however no other tools are added. Using the selection manager the tool, in this case the tool including PROMT, acquires a reference to the currently selected object and manipulates it via the manipulation manager. The messenger pattern is used for communication between the separate parts.

As seen in figure 5.3, the green objects are based on the *VRTK* framework and the blue ones are created in this project. The *Controllers* are granting a layer of abstraction between hardware controller input and events fired by these. The *Tool Manager* handles different tools such as the *Prompt Tool* developed during the thesis project, which themselves contain different interaction techniques to form a functioning tool. There is a differentiation between gripping an object, which means directly manipulating it by touching it and moving it with the controller, and selecting an object which then allows interaction via a chosen tool. The way in which an object is selected is determined by the current tool used. In addition, a *Snapgrid* singleton is used by the *Sketch Object* to snap its transformation to a chosen point, scale, or rotation.

5.3 Main Parts

This section describes the most relevant parts of the application in detail. The described parts are entities and classes of the application. The section will focus on the parts of the application that are involved in the design of PROMT rather than the whole use case application.

5.3.1 Primitives

The primitives, or *Sketch Objects*, are the objects that are manipulated in the use case application. They are comprised of different three-dimensional

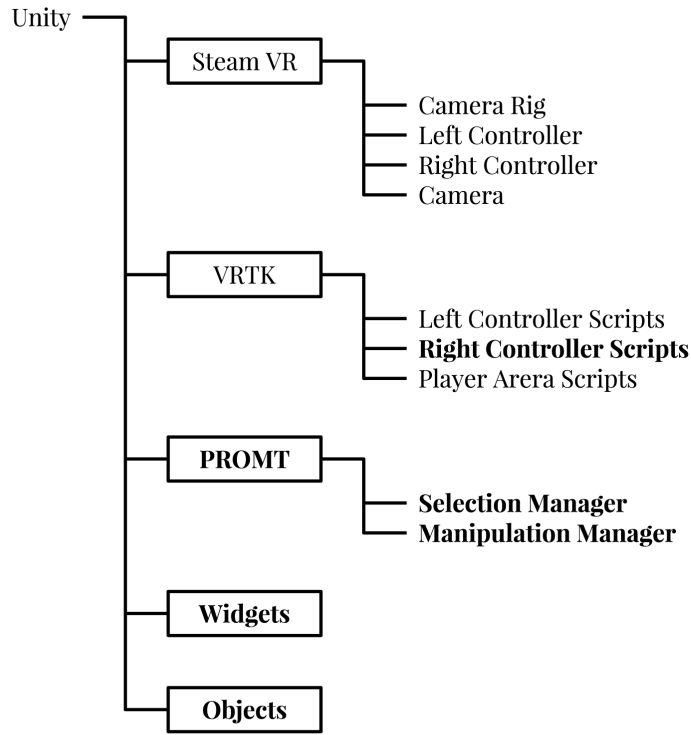


Figure 5.2: An abstract representation of the *Unity* project, showing the parts created for the thesis project in bold letters.

geometric primitives. These primitives only differ in terms of their appearance and the shape of their colliders. They have very little functionality on their own, apart from knowing that they are primitives and holding a flag to show if they are selected or not.

5.3.2 Widgets

The widgets in this application do not hold functionality to actually manipulate an object. They serve mainly as a visual aid for the user. Therefore the widgets need a corresponding tool, as described in section 5.3.5. The widgets functionality is limited to adapting their appearance to the currently selected object and the current manipulation settings of the application. The widget shown on a selected object for example adapts its orientation and size depending on the object it is shown on at the moment. It also adapts to the currently selected axis, by changing colors, and to the currently selected manipulation mode, by changing shape. The widgets are closely connected

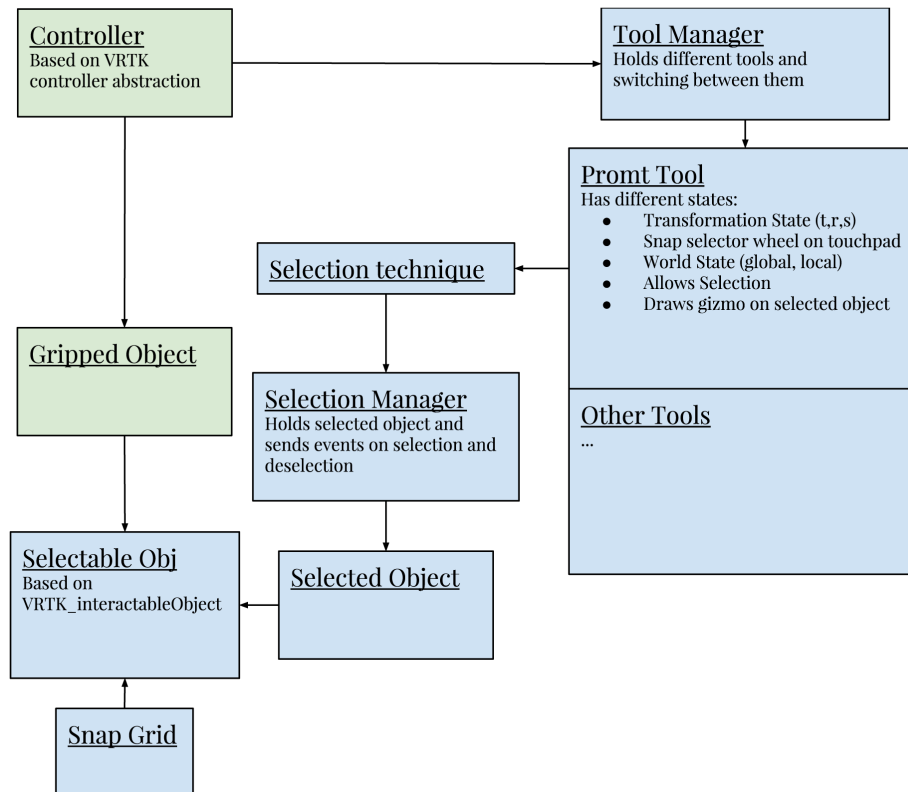


Figure 5.3: A simplified view of the project architecture.

to the tool used, as well as the selection manager described in section 5.3.6.

5.3.3 Controller

The controller is the virtual representation of the physical controller held by the user. Its main function is to receive input of the physical controller and sending it forward. This is done while adding a layer of abstraction to allow using different physical controllers without adapting the virtual representation. For this specific application, the controller also holds the so-called tool manager, which is used to handle multiple different tools. These tools are also contained in the controller entity.

5.3.4 Tool Manager

The tool manager is used to hold a list of all tools and switch between them. It is modeled after the state pattern. The tool manager receives messages from the controller and sends them only to the currently selected tool, which

then acts accordingly. In this application, the tool manager was implemented as a precaution in the case that switching between multiple tools during runtime should be necessary.

5.3.5 PROMT Tool

Tools are the scripts containing the functionality to manipulate objects or potentially execute other functions. In this application, multiple tools can be implemented and then handled using the tool manager. The controller input sent by the tool manager is received by the currently selected tool which then acts accordingly. The PROMT tool uses the interaction design described in chapter 4 and applies the user input to the selected object.

5.3.6 Managers

There are three managers in the application. These managers make use of the singleton pattern to ensure there are not multiple instances of them. They are generally used to hold data that is used by multiple different parts of the application.

Snap Grid Manager

The snap grid manager only holds information over the currently set size of the snapping grid as well as the rotational step size. As the grid is not actually stored in memory, it does not hold data of the grid points. The snap grid functions by calculating the next possible position in a potential grid at given grid size.

Selection Manager

The selection manager is used to hold the data needed to ensure the right object is manipulated in the right way. It holds the currently selected object, which the manipulation will be applied to. It also holds information about the selected axis, if any is selected, and the manipulation mode. Other than holding this information, it also informs other entities if this data changes.

Manipulation Manager

The manipulation manager holds data about the position, rotation, and scale of the controller. It also holds the selected object and calculates differences between its transformation and the controller's transformation. It is also used to calculate the difference in position, rotation, and scale over a certain time.

After describing the implementation of the application and its main parts, the following chapter will focus on the evaluation of its features. Af-

ter describing the methodology and the technical setup the results will be shown. This is followed by an analysis of these results.

Chapter 6

Evaluation

The last chapters described the concept and implementation of an improved object manipulation technique based on existing research in the field of user interaction in immersive virtual environments. To appraise the feasibility of the developed technique described in chapters 4 and 5, a series of expert interviews have been conducted.

This chapter will examine the methodology used to conduct the evaluation. Also it will describe the hardware used in the evaluation as well as the setting. In the last section, the results of the evaluation are presented.

The purpose of this evaluation is to analyze if the expected improvement of precision during object manipulation was successful. Also other aspects of the interaction were examined such as the use of haptic feedback, menu items, and the selection of objects.

6.1 Methodology

The evaluation will be conducted using the *Thinking Aloud* method after Jakob Nielsen [15] using experts of the fields usability and virtual environments as test subjects. The *Thinking Aloud* method was chosen for its ability to convey what the user really thinks about a design [24]. Three users will separately receive an explanation of the technique and the application, followed by them using the application. After the users tested the application, they were asked a set of questions and afterwards for their personal feedback.

The three experts from the academic staff of the University of Applied Sciences in Hagenberg that evaluated this project are: Dr. Christoph Anthes, who has conducted extensive research about virtual and augmented reality technology, as well as interaction techniques in these fields and is now teaching a virtual reality course; Michael Lankes, who is researching and teaching in the fields of user interaction, user experience, and usability and interaction design; the third expert is Hans-Christian Jetter, who is a professor for user experience and interaction design and is also researching

and teaching in human centered computing. An interview with three experts to evaluate the project was chosen to achieve quick results and receive a first, quick evaluation to find major problems within the project. Each interview was held separately. During tasks the users were allowed to ask questions and suggestions were made to improve interaction.

6.2 Setting

The interview was held in a room with a *HTC vive* virtual reality system set up. The tracking area was limited to approximately two by two and a half meters. For the interview only the tester and an interviewer were present in the room.

First each expert was given an introduction to the project. This introduction included an explanation of the the projects concept and goal, also the use case was explained and a quick overview of the button layout was given. In the next step, the interviewer put on the head-mounted display and explained all the features to each expert while they watched the process on a screen. During this passive learning part, the experts were introduced to the button layout of the controllers, the pointer and the grip button, the transformation settings, the close range interaction, the axis alignment interaction, and the menu interaction. This was followed by them putting on the headset and trying out all these points themselves, while getting familiar with the application.

The introduction was followed by three tasks. Each task was aimed at a different way of interacting with objects using PROMT. The first task was to set a table. The user was given a set of primitives depicting dishes and food items and a fixed table was placed within reach of the user. The focus of this task was to interact with objects at a close range. The users were told to focus on placing the objects precisely and try to avoid intersections between the objects.

The second task was to build a wall with a window in it. The wall was specified to be exactly two meters high and the window in the wall should have a size of 50 by 50 centimeters. All other parameters were undefined. At the beginning of the task, the user was given a wall panel that was one by one meter in size. This task was made to see how the experts would handle interaction with large objects at close range. Also using numerical feedback and discrete manipulation in combination with snapping to achieve precise dimensions was part of this task.

The third task was to assemble a large snowman. The parts of the snowman were placed outside of the user's reach. The incentive for this task was to see how users would handle interaction with remote objects and to force them to use indirect interaction. Also the scaled translation was meant to be used in this task, for moving objects over long distances by quickly moving

the hand.

6.3 Results

The results of the expert interviews conducted as described in the previous sections are now presented in the following section. These results are based on the comments made by the evaluators during the testing phase, as well as their answers during the questioning phase. The data from these two phases is not separated as they include feedback concerning the same topics. To summarize the data gained during the evaluation, they are clustered in a few main topics depicting the most prominent features discussed.

6.3.1 Hardware Interface

The *HTC Vive* controller, as the main input device, served as the hardware interface for this project and was used in the expert interviews. The two buttons used mainly for selection and manipulation have found to be the biggest obstacle during interaction. The trigger button located at the user's index finger was mainly used to select objects via ray casting. The grip button, located at the user's middle finger and activated by squeezing controller, was used to manipulate object, as it resembles gripping an object. The evaluators had problems differentiating between which of those buttons should be used to activate either interaction. Mainly the grip button was problematic, as its activation was found to be unintuitive. It was also suggested to map selection and manipulation on the same button. Other than that, the evaluators had no problems using the controllers.

6.3.2 Manipulation Metaphors

Translating objects along a certain axis was easy to use for all evaluators; the scaled input was more complicated to use. It was very subtle, so it sometimes was misjudged as an unintended error. Also the scaled-down translation on slow movements was hardly used. Mainly it was used to quickly move objects over larger distances.

The scaling metaphor was not always clear to all evaluators. In general it was used as intended; the fact that objects were scaled down when the controller is moved towards the object and scaled up when the controller moved away was not clear. Also accidental scaling of objects occurred more often than other errors during manipulation.

The crank metaphor used for rotating object along a certain axis was shown to be understandable. Some problems occurred during its use nonetheless. Mainly, compared to the other metaphors, it was more complex to use. The widget provided during use was placed on the position of the controller. This led to evaluators focusing on the object and therefore missing feedback

given by the crank-widget. Also it required more effort to use; the arching motion users have to mimic made interaction slightly exhausting. This also led to a more time-intensive interaction. The precision gained through increased distance to the handles center was noticeable, and it was preferred over rotating the hand which produced strain on the user's wrist.

The haptic feedback given when manipulating objects along a single axis was perceived in different ways. On one hand, the feed back was too frequent, which had the unintended effect of not giving the user clear information. On the other hand, it was not noticeable enough, also resulting in a lack of information.

6.3.3 Axis Alignment

Axis alignment for degrees of freedom separation was adopted well by all evaluators. The learning time differed between individuals. After the concept was clear, however, all evaluators found it to be useful for manipulating separate axis of an object. The connection of hand movement and the movement of the manipulated object was mentioned to be intuitive. Axis alignment was mainly used for large or distant objects by the evaluators. For small objects within reach, direct manipulation was preferred; also if a high degree of precision was not necessary, axis separation was hardly used. In general, axis alignment was not perceived as complicated by any evaluator.

6.3.4 Selection

The ray casting selection, as well as the selection by touching objects, was well received in general. It was easily usable and intuitive. The ray casting however has shown deficits when selecting small objects at larger distances because it heavily relied on the user's precision. Also the testers would have preferred a more noticeable feedback when hovering over a selectable object.

6.3.5 System Controls

The menu shelf did not pose a problem to the evaluators. Only the positioning of the menu when called was slightly unclear. Also the selection of the different manipulation methods was mainly understandable. It was apparent that the position of each method had to be memorized before interaction was fluent but it has shown to be only a small distraction from the actual task.

6.3.6 Grid Snapping

The grid snapping, as well as the discrete manipulation was only used in special situations, such as the placement of wall panels that were all placed in the same orientation at constant distances. When used it was well received.

The setting of parameters on the menu was not very intuitive and the fact that discrete manipulation along an axis could result in an object leaving the global grid was confusing.

6.4 Analysis

The expert interviews have pointed out aspects of the technique, as well as the application, which can be further improved to create a more user friendly and efficient interaction. Certain aspects of PROMT such as the crank handle metaphor need improvements in some details, others like the translation and degrees of freedom separation through axis alignment have been implemented and used as intended. The following analysis focuses on aspects of PROMT that have been developed during the implementation of the project. Aspects included in the use case application that have been discussed in literature before will not be analyzed.

- The concept of degrees of freedom separation and its implementation via **axis alignment** have proven to be useful and easy to learn. To further improve axis alignment and make it even more self-explanatory, certain steps could be taken. The feedback users receive can be adjusted to improve performance for new users. An enhanced and more noticeable haptic feedback could make it easier for users to realize when an axis is selected and deselected. Also improved visual feedback on the object could help to make it easier for users to understand how they have to orient the controller to select an axis.
- The complexity of the **crank handle** metaphor has led to a time costly interaction. While the basic concept may be feasible, the current implementation seems to be too complex. The visual feedback, delivered by the widget on the user's hand, missed its purpose as users mainly focus on the object they interact with and not on visual feedback located on the position of their hands. Therefore improvements could be made by changing the position of the visual feedback to be inside the user's view. Also the threshold between the beginning of the crank handle interaction and the actual rotation could be adjusted to allow quick rotations with smaller movements.
- **Scaled translation**, comparable to the PRISM technique (see section 3.9), has aided users in quickly moving objects over large distances. The scaled-down manipulation was however mainly ignored, or completely missed. To improve the scaled-up motion, it might be advantageous to not only consider the speed of the user's hand, but also the distance of the object. This could make translations close to the user as well as translations of remote objects more uniform and easier to use. To improve the use of scaled-down motion, the user might have to be informed about the current state using some form

of feedback. One possibility would be to give haptic feedback which adjusts its frequency to the current motion scale. Another way would be to find a form of visual feedback demonstrating the current speed. This could be done by visualizing movement vectors of the hand that change in color or size depending on the current scale of motion.

- The main problem of **grid snapping** was the disparity between moving an object in certain axis and moving an object in all axis freely. When moving an object freely it snaps to a global grid, otherwise it snaps to the local grid of the current object. To clarify these states, it might be beneficial to treat those forms of interaction more separately. Visualizing the global grid and setting its size in a separate location might clarify that it is not connected to the discrete manipulation of an object. This way problems like users not being able to place objects precisely when manipulating them discretely, because of the objects not being placed in the global grid, might be minimized.

The conducted evaluation and its analysis have shown that PROMT has potential to increase precision during object manipulation in immersive virtual environments. The problems that occurred during evaluation can be addressed without changing the main concept of the techniques design. After this analysis the following chapter will provide a conclusion of the thesis and point out possible future work.

Chapter 7

Conclusion

7.1 Summary

This thesis presented the design and development of an interaction technique aimed to improve precision of object manipulation tasks in immersive virtual environments. Based on knowledge gained from existing research in the field of interaction design in virtual environments and in consideration of existing interaction techniques, a design for PROMT, a precise, remote object manipulation technique has been developed. This technique's design, as the name suggests, was developed with the goal of creating a technique that improves upon existing techniques in terms of precision while maintaining quick interaction. Unlike many techniques for immersive virtual environments, it does not try to mimic natural interaction in the physical world. Instead it focuses on making up for lost precision because of missing numerical input by utilizing the six degrees of freedom provided by tracked controllers.

In addition the design of an application, used as use case for PROMT, was presented. This use case is an application aimed at being used by industrial designers, to quickly create mock-ups of three-dimensional designs and evaluate them in terms of proportion and shape. This evaluation is especially interesting for designs that are too large to fit on a computer monitor in life-size, but within room scale.

The design process resulted in a technique combining degrees of freedom separation, gestural input and three-dimensional widgets. The separation of degrees of freedom has been suggested before as a method of increasing precision in manipulation tasks. Combined with the gestural input of aligning the controller with an axis to select it, and simple metaphors to subsequently transform the object in said axis, PROMT has shown itself to be a feasible technique.

The conducted evaluation in form of expert interviews have suggested that PROMT has potential as an object manipulation tool for industrial

applications. Problems with the crank handle metaphor are still a subject to be addressed in future research, as they have not shown similarly good performance during tests. Also features of the application designed as use case have shown again that state-of-the-art interaction techniques are not yet at a point of best practice approaches.

7.2 Future Work

As virtual reality technology is at a point where it can be used reliably and hardware is affordable, the use in industrial application for three-dimensional design is becoming an interesting field of application. Moving the user in the same space as his or her creation does however also pose problems. New interaction techniques have to be found to get interaction in immersive virtual environments to a point where the user experience and the user performance are comparable to traditional desktop-based techniques, and possibly surpass them.

Just as PROMT has shown potential to bring virtual reality interaction to that point, it has also shown that it still needs improvement. Besides refining the core feature of axis alignment, it would be suggested that improvements to the separate manipulation metaphors could significantly improve the performance of the technique. Especially the task of rotating objects precisely is a point to improve. While the crank handle metaphor has potential it still lacks the needed performance because of its complex interaction.

A follow-up project could also address the limited features of the use case application. Because the main focus was set on the interaction technique for object manipulation, it was decided to create a use case with a small set of features. However there is potential to improve this application in different ways. The evaluation has also suggested that PROMT has more potential when being used as sole mean of basic object manipulation to reduce complexity.

One feature that would be very interesting to explore would be possibilities to improve the object selection, for instance, by addressing the problem of selecting occluded objects. Also ways of selecting multiple objects could improve interaction in similar applications. While ray casting is a very intuitive form of selecting objects, it has shown a lack of precision, based on the user's own inability to be perfectly precise. Finding a way to improve ray casting by reducing the user introduced imprecision could render this selection technique more viable.

Another interesting feature that could be explored further is context-sensitive data representation in immersive virtual environments. Industrial applications could benefit greatly from improved representation of relevant data at crucial points of manipulation. The user interaction in virtual im-

mersive environments is a field of research that offers a lot of possibilities for finding new exciting forms of interacting with computers and virtual environments. The possibilities for innovation are manifold and have the potential of inspiring researchers for some time to come.

Appendix A

Protocol from the Expert Interview

I: Interviewer

T1: Christoph Anthes

T2: Michael Lankes

T3: Hans-Christian Jetter

A.1 Interview 1

I: Bei dem ersten Task hast du einen Tisch vor dir, einen grauen Tisch, den kannst du nicht manipulieren und links daneben hast du ein paar Primitive. Diese sollen ein Glas, einen Teller und drei Essensgegenstände darstellen. Der erste Task ist es diese auf dem Tisch zu platzieren, so wie du es für richtig hältst den Tisch zu decken und darauf zu achten, dass die Objekte wirklich auf dem Tisch stehen und so gut wie möglich keine Überschneidungen stattfinden.

T1: So gut wie möglich, okay.

I: Ja. Also du kannst beschließen wann du fertig bist mit dem Taks.

T1: Okay, aber den Tisch kann ich nicht selektieren.

I: Genau.

T1: Gibt es da eine Möglichkeit diese wieder zu re-alignen, also sie in die Ursprungsposition zu bringen oder etwas in der Art?

I: An die Ursprungsposition nicht, aber du kannst die Rotationsschritte auf 45 Grad beschränken dann bekommst du es relative leicht wieder in eine Ausgangsposition.

T1: Weil das hab ich mir natürlich hier jetzt irgendwie vermurkst in dem ich es immer frei bewegt habe.

I: In dem Fall wäre es wahrscheinlich am einfachsten die Rotation zu beschränken und dann das Objekt grundsätzlich wieder auszurichten. Dann

kann man es im Grunde wieder frei bewegen, ziemlich schnell.

T1: Okay, ich hätte mal gesagt das passt so. Da ist die Frage wie genau man da werden möchte. Es ist natürlich schwierig wenn man kein snap-tool hat. Okay, passt.

I: Dann fangen wir mit dem nächsten Task an. Bei diesem Task hast du jetzt nur dieses Wandpaneel links neben dir stehen. Mit diesem Wandpaneel kannst du jetzt vor dir eine Wand aufbauen. Du kannst diese Panel skalieren und kopieren und deine Aufgabe ist es, in diese Wand ein 50 mal 50 Zentimeter großes Fenster einzubauen und die Wand soll zwei Meter hoch sein.

T1: Zwei Meter hoch und 50 mal 50 groß, na servas. Das ist jetzt drei Meter?

I: Nein das sind jetzt 10 Zentimeter. Es ist etwas irreführend, das ist der dritte Einstellungspunkt.

T1: Die Dimension sehe ich da? Translation, Skalierung, Meter mal Meter. Okay, das heißt jetzt muss ich eigentlich die ... Die Wand Tiefe ist egal oder?

I: Die Wand tiefe ist egal.

T1: Ist es egal wo ich die Wand hinstelle?

I: Die kann einfach da stehen.

T1: Auf welcher Höhe soll das Fenster sein? Gibt es da irgendwelche Präferenzen?

I: Das ist dir überlassen wo sich das Fenster befindet.

T1: Zwei mal zwei hast du gesagt?

I: Zwei Meter hoch, die Breite ist auch dir überlassen.

T1: Das ist jetzt etwas blöd da ich nicht in demselben Grid bin.

I: Wenn du den Würfel in der Mitte berührst dann snapst das Objekt zu einem globalen Grid. Aber die anderen Objekte sind wahrscheinlich nicht auf diesem Grid.

T1: Das kann durchaus sein. Dann war das das Problem, verstehe schon. Okay dann sollte das jetzt passen. Wir haben zwei Meter Gesamthöhe und auf einem Meter Höhe...

I: Der dritte Task zielt darauf ab, dass du nur auf Distanz manipulierst. Das Objekt das du jetzt manipulierst wird jetzt außerhalb von deinem Bewegungsbereich sein und es stellt einen riesigen Schneemann dar. Deine Aufgabe ist es diesen Schneemann zusammen zu bauen. Du hast keine Möglichkeit direkt dorthin zu gehen.

T1: Aber ich kann bis zu meine Trackingrand gehen, das ist natürlich besser und ich kann auf eine geringere Distanz noch besser interagieren, logischerweise. Wie präzise der jetzt zusammen gebaut ist, das kann schon eigenes Ermessen sein, oder?

I: Es sollte als Schneemann erkennbar sein, aber es ist keine gewisse position von Objekten erforderlich.

T1: Die Distanz ist eher das Problem (beim Selektieren, Anm. d. Verf.).

So jetzt mache ich das, was mir noch ein bisschen fehlt, ich drehe nämlich jetzt, beziehungsweise ich bringe dieses Zeug zu mir. Passt das so für die Nase? Und die Kollegen hier? Werden das Knöpfe oder Augen?

I: Das werden zwei Augen und drei Knöpfe.

T1: Okay. Ich habe schon lange keine Schneemänner mehr gebaut. Okay. Das was mir etwas fehlt, ist der Treffer beim Selektieren. Ich würde gerne sehen wenn ich darauf pointe, dass mein Collision-Point ein pickable Objekt erwischt. Also ich sehe schon eine Kollision an dem das ganze anstößt, sprich wenn ich jetzt auf eine Wand oder ähnliches zeige, sehe ich den Kollisionspunkt, was gut ist. Aber ich sehe nicht, ob das jetzt tatsächlich das Objekt ist, das ich auch bewegen kann. Also ich könnte genauso gut am Boden landen. Das wäre so eine Sache die ich persönlich ganz nützlich fände. Ansonsten ist es immer noch viel Armarbeit. Das geht wahrscheinlich über diese Geschwindigkeits-Geschichte noch etwas besser, aber das ist ... Es sind übrigens vier Knöpfe.

I: Stimmt das war ein versteckter Knopf. Gut das wars dann. Dann würde ich dir jetzt noch ein paar Fragen stellen. Die erste Frage ist, inwiefern du das allignen des Controllers mit der Achse sinnvoll empfunden hast. Inwiefern das für dich ein praktische Art war die Achsen zu selektieren.

T1: Hängt davon ab was du damit machst. Es gabe Sachen bei denen es Sinn gemacht hat; es gibt Sachen bei denen es weniger Sinn gemacht hat. Ich fand es zum Beispiel extrem hilfreich bei diesem Wand zusammenbauen, weil du da auch einen Grid hast an den du es Snapst und die Teile sind aligned so wie du sie gern hättest, das heißt du kannst direkt einen Eindruck bekommen, welche Achse du wie haben willst. Du kannst damit, finde ich persönlich, extrem schnell auswählen in welche Richtung du verschieben willst. Und das ist schneller als die Achse an zu klicken. Wo es allerdings, finde ich, wenig bringt ist bei der ... habe ich zumindest den Eindruck, es kann sein dass ich mich jetzt täusche ... bei der Geschichte mit diesem Schneemann habe ich mir gedacht, du bist da etwas freier unterwegs, ich glaube da habe ich weniger mit den Achsen gearbeitet. Und bei dem ersten Task habe ich eigentlich fast gar nicht mit den Achsen gearbeitet. Aber bei dem Alignen der Wandteile fand ich es sehr hilfreich und ich gehe auch davon aus, dass es anders länger gedauert hätte. Also ich finde es macht dann Sinn wenn ich das einfach mappen kann. Also das Bild, ich weiß, ich will die Ausrichtung haben und kann das einfach mappen. Wo es jetzt dadurch, dass das snapping eingestellt war eine eingeschränkte Umgebung war und dadurch auch darin recht gut funktioniert hat. Das ist meine Interpretation.

I: Okay. Was war für dich das komplizierteste an der Sache mit dem Allignen? Was war daran vielleicht unverständlich?

T1: Bei dem Allignen fand ich eigentlich wenig unverständlich. Es gibt ein paar sachen an dem Interface die ich weniger intuitiv als das allignen finde. Zum Beispiel die Verwendung von dem Grip-Button. Das kann sein, dass das eine persönliche Aversion von mir gegen den Knopf ist, nicht nur

speziell in dieser Anwendung. Ich nutze ihn selbst auch gar nicht weil ich ihn einfach vollkommen unintuitiv finde, was jetzt nichts mit dem Interface zu tun hat, sondern einfach mit dem Controller.

I: Das haben schon mehrere Leute bemerkt und ich bin auch kein großer Fan von dem Button.

T1: Der ist schwierig. Das Alignen an sich, finde ich, funktioniert gut. Was man da vielleicht machen könnte, was ich generell von der Feedback-Geschichte her vielleicht nicht so geschickt finde ist, was du machst, ist Feedback zu geben, wenn du etwas translierst. Du gibst relativ viel Feedback, du willst aber eigentlich nicht so viel Feedback haben, sondern du willst wenn du irgendetwas relevantes machst Feedback haben. Das heißt, wenn ich zum Beispiel irgendetwas zu manipulation selektiere, oder auch wenn ich eine Achse auswählen zum allignen. Dann würde ich tendenziell ein Force-Feedback geben, nicht wenn ich etwas verschiebe. Das wäre so eine Sache bei der ich mir denke, damit könnte man das Alignen noch intuitiver gestalten weil du dann ja teilweise noch nicht einmal hinschauen müsstest, wenn du das machst.

I: Also beim Allignen gibt es ein Feedback. Aber die Feedbacks sind generell sehr minimal eingestellt, weil schon bemängelt wurde, dass diese zu extrem sind.

T1: Okay. Na ich weiß nicht. Ich fand es da vielleicht etwas zu subtil, im Vergleich zu dem Feedback, dass du sonst hast.

I: Wie hat dich die skalierte Bewegung bei der Translation beeinflusst, hast du das positiv oder negativ bemerkt?

T1: Ganz am Anfang habe ich es für einen Bug gehalten, ich weiß nicht ob du mir gesagt hast, dass es skaliert. Da kam es mir komisch vor. Ich hab es dann versucht es ein bisschen bei dieser Schneemann-Geschichte einzusetzen. Es macht ja keinen Sinn, wenn du es auf kurze Distanzen machst. Bei dem ersten Task kannst du ja gar nichts machen in der Richtung, weil es da im Prinzip ja um Präzision geht.

I: Es skaliert die Bewegung aber auch herunter. Also wenn du deine Hand langsam bewegst, bewegt sich das Objekt noch langsamer.

T1: Okay, da habe ich jetzt gar nicht daran gedacht. Ich hätte gedacht, dass es von der Geschwindigkeit her nur darüber geht. Das ist richtig, dann macht es in dem ersten Task wahrscheinlich auch mehr Sinn. Das hätte ich jetzt nicht erwartet, sonst hätte ich es da vielleicht auch versucht. Bei dem Snapping hab ich es eigentlich nicht gebraucht. Und bei dem letzten (Task, Anm. d. Verf.), nachdem man da über größere Distanzen arbeitet, fand ich es da schon irgendwie sinnvoll, aber es ist schwer es zu kontrollieren finde ich, vom Interface her. Dafür musst du dich glaube ich schon sehr, sehr lange damit beschäftigt haben, um da ein Gespür zu haben mit welcher Geschwindigkeit du Sachen hin und her schleuderst.

I: Würdest du sagen, dass man da irgendeine Art visuelles Feedback, oder eine andere Art von Feedback hinzufügen könnte, damit das sinnvoller

für den Benutzer erkennbar ist?

T1: Ich weiß es nicht. Wenn man mit Geschwindigkeiten arbeitet, was man da gerne als Feedback einblenden kann, sind Vektoren. Du kannst also vom Prinzip her Streifen zeichnen, wo sich das Objekt bewegt hat, und dann da beispielsweise die Farbe ändern oder die Größe ändern je nach Geschwindigkeit. Das wäre eine Möglichkeit, aber das Problem, dass ich bei der Sache ein bisschen sehe ist, weil ja um das vernünftig zu beschleunigen musst du dich ja auch richtig schnell bewegen. Dann weiß ich nicht, ob du so auf das visuelle Feedback dann auch achtest. Das ist ein bisschen schwierig. Grundsätzlich wäre feedback sicher gut, aber ich weiß nicht ob es speziell in dem Fall wirklich sinnvoll ist. Das müsste man sich anschauen.

I: Was ist deine Meinung bezüglich der Kurbelbewegung, um entlang einer Achse zu rotieren?

T1: Es hat Vor- und Nachteile. Der Vorteil ist, ich habe den Eindruck es ist präzise, ich habe den Eindruck ich kann es gut kontrollieren und es ist intuitiv. Der Nachteil ist es dauert eine Zeit lang. Also es ist aufwändig und du hast eine relativ große Bewegung die du auch schlussendlich ausführen musst. Das finde ich ein bisschen schwierig. Mir gefällt es aber besser, als diese Drehbewegung an dem Button. Die hatten wir ja im direkten Vergleich, bei dem ich mich auch gefragt habe, warum ihr nicht das selbe Interface verwendet, aber wahrscheinlich um die miteinander zu vergleichen, hätte ich jetzt vermutet, aber ich weiß es nicht. Aber das Problem dass du da natürlich hast ist, dass du mit der Drehung eingeschränkt bist, das ist schon klar. Da funktioniert das Kurbeln wesentlich besser, aber es hat auch einen gewissen Aufwand den es mitbringt.

I: Wir haben zwar schon kurz darüber geredet, aber wie befriedigend war diese haptische Feedback für dich, wenn du ein Objekt manipuliert hast entlang einer Achse?

T1: Wenn ich kein snapping aktiviert habe dann ist es mir zuviel und wenn ich snapping aktiviert habe, dann weiß ich auch nicht, ob ich es an dieser Stelle brauche. Ich hätte es lieber an einer anderen Stelle gehabt. Zum Beispiel bei der Selektion oder auch, was zum Beispiel bei dem letzten Task interessant gewesen wäre, wenn ich einen validen Hit-Point getroffen hätte. Also wenn du so eine Kugel auf Distanz selektierst und es dann haptisches Feedback gibt, dass dir sagt ich kann das Ding jetzt aufheben. Aber ich hab da weder visuelles noch haptisches Feedback, das macht es noch ein bisschen schwierig. Da hätte ich es sinnvoll gefunden. Beim Snapping brauch ich es vielleicht, da kann ich es mir noch vorstellen weil ich dann in Schritten arbeite. Aber wenn ich mich kontinuierlich bewegen muss ich kein Feedback bekommen um zu wissen, dass ich meinen Arm bewege.

I: Im Grunde ist, wenn der Snapping-Grid auf die kleinstmögliche Einstellung gestellt ist, die Bewegung immer noch nicht kontinuierlich, sondern diskret. Das heißt, das Feedback das man bekommt ist jeden Millimeter ein haptisches Feedback.

T1: Das ist richtig, aber wenn ich ein kontinuierliche Bewegung habe, habe ich damit natürlich eine gewisse Frequenz die sich als Rumble anfühlt. Dort brauche ich es nicht, da irritiert es mich mehr in dem Fall.

I: Also eher wenn die Schritte größer sind, um wirklich zu bemerken wenn ein Schritt passiert ist?

T1: Genau, es macht dann Sinn, finde ich, wenn ich irgendeinen Mehrwert, eine Zusatzinformation bekomme. Zum Beispiel, wenn ich an eine Grid-Punkt andocke. Wenn das Grid so gering ist, dass ich das weder visuell noch anders differenzieren kann, dann brauche ich nicht permanent Feedback zu geben, dass sowas passiert. Dass ist eigentlich reziprok proportional, ich habe viel Feedback für wenig das ich gemacht habe. Und ich brauche eigentlich Feedback an der Stelle an der es relevant ist.

I: Wenn man mit relativ kleinen Objekten, in einem kleinen Bereich arbeitet ...

T1: ... kann es sinn machen.

I: Dann hat man wahrscheinlich wieder den selben Effekt als würden man mit einem größeren Objekt in einem größeren Grid arbeiten.

T1: Vielleicht, wobei dann auch wieder die Frage besteht, wie es sich mit der Präzision verhält. Weil du dich dann, wenn du dich relativ langsam bewegst, wahrscheinlich auch ständig Feedback bekommst. Also wenn du sagen würdest ich muss das jetzt Millimeter genau irgendwo anlegen, wenn du so einen Task hast bei dem es wirklich auch visuell sichtbar ist, dass es auf den Pixel passen muss, dann kann es da vielleicht Sinn machen. Aber ich glaube ich würde nicht im Millimeterbereich operieren wollen, sondern eher im Zentimeter Bereich im Normalfall.

I: Welche Probleme hast du beim Ray-Cast selektieren von Objekten gehabt?

T1: Naja, die klassischen Sachen, es ist auf Distanz nicht einfach. Da fand ich war das Hauptproblem das fehlende Feedback, also es ist nicht so, dass gar keines da ist. Ich sehe den Ray, da habe ich schon einmal Feedback, ich sehe den Kollisionspunkt, allerdings mit allem. Aber da hätte ich mir wirklich Feedback in dem Punkt gewünscht, dass ich den Kollisionspunkt mit Objekten die ich tatsächlich bewegen kann, visuell dargestellt bekomme. Da kann man vorne den Knubbel vielleicht verändern oder den ganzen Ray ändern. Je nach dem wie intensiv man das darstellen möchte. Auf der anderen Seite hätte ich mich auch über Vibrationsfeedback gefreut. Da muss man aber auch etwas damit aufpassen, wenn man so etwas implementiert. Es kann auch extrem störend sein, wenn du 50 Kugeln nebeneinander liegen hast und du gehst da drüber und das ganze vibriert auch nur ständig. Da muss man natürlich erst mal prüfen, ob ich auf einem neuen Objekt bin und wie oft ich wechsele. Da kann man eine ganze Menge machen um das geschickt zu machen. Das hat mich bei dem Ray-Cast am meisten gestört, dass ich kein Feedback hatte.

I: Was war die größte Herausforderung, wenn du Objekte selektiert hast?

Du hast ja nicht nur mittels Ray-Casting sondern auch durch Berührung Objekte selektieren können. Was war für dich dabei das größte Problem. Oder war das das Selbe?

T1: Ich glaube das war am Anfang . . . nein, also ich mein das war klar, bei dem Ray-Cast ist natürlich die Präzision ein Problem, aber ich habe ein größere Reichweite. Das wiegt man dann ein bisschen ab. Was ich am Anfang schwierig fand ist, dass du doch verschiedenen Möglichkeiten hast zu interagieren. Das ist etwas, dass zuerst etwas erlernt werden muss, da du doch auch viele Optionen anbietest, auch zeitgleich und mit unterschiedlichen Buttons. Das sind jetzt eigentlich nur noch zwei Buttons, der Grip und der Trigger. Aber vielleicht wäre es da geschickter es uniformer zu gestalten. Aber da weiß ich jetzt auch nicht genau wie ich das machen würde.

I: Im Sinne nur eine Möglichkeit anzubieten etwas zu selektieren?

T1: Nein im Sinne nur einen Button für beide Interaktionen zu verwenden und dann vorzugsweise den Trigger. Ich kann grundsätzlich sagen ich ray-caste immer, es sei denn ich habe eine Kollision. Dann ray-caste ich quasi immer, aber eine Kollision kann ich trotzdem behandeln. Warum differenziert ihr?

I: Der Grund warum ich zwischen den beiden Buttons differenziere ist . . . wenn du ein Objekt mittels Axis-Alignment manipulieren möchtest, dann kann der Controller in eine beliebige Richtung zeigen, während sich das Objekt dort drüben bewegt. Wenn an dieser Stelle dann ein anderes Objekt ist, ist es schwer heraus zu finden, ob ich dieses Objekt selektieren möchte oder ob ich manipulieren möchte. Deswegen habe ich mich dazu entschlossen, einen gewissen Button zu haben mit dem ich immer selektieren kann und einen mit dem ich immer manipulieren kann. Damit da keine Missverständnisse passieren.

T1: In dem Kontext ist es natürlich logisch.

I: Vielleicht würde sich das Problem lösen, wenn die Controller angenehmere Buttons haben, oder sinnvoller Buttons als diesen Grip-Button.

T1: Was vielleicht gehen könnte, aber das weiß ich jetzt auch nicht . . . Du hast verschieden Stufen, den Trigger kannst du auch abfragen, ob der ganz durchgezogen ist oder nur gedrückt ist. Im Moment feuerst du den Ray beim Andrücken raus.

I: Sobald er leicht berührt wird.

T1: Genau beim Berühren. Und du versuchst zu selektieren, wenn der Button ganz gedrückt ist. Jetzt ist die Frage, was wäre wenn der Ray die ganze Zeit aktiv wäre, was auch problematisch sein könnte, sodass ich bei einer leichten Berührung selektiere und beim Durchdrücken manipulierte?

I: Da wäre das Problem, dass du in der Zeit in der du ein Objekt selektieren möchtest eventuell bereits das zuvor selektierte Objekt manipulierst. Vielleicht kann man da auch noch eine besser Möglichkeit finden. Okay machen wir weiter. Wie sehr hat es dich abgelenkt, zwischen den Manipulationsmöglichkeiten auszuwählen, also rotieren, translieren und skalieren? Hast

du dabei den Fokus von der Aufgabe nehmen müssen, um eine Manipulation auszuwählen?

T1: Logischerweise schon, klar. Aber das fand ich nicht so tragisch. Es wirft einen zwar ein Stück raus, aber ich glaube die Vielfalt der Optionen im Allgemeinen ist das größere Problem. Da müsste man sich auch die Tasks anschauen, denn da wechselst du nicht viel. Die meisten der momentanen Tasks sind prinzipiell Translationsaufgaben. Ich glaube es würde noch mehr auffallen, wenn man öfter dazwischen wechseln müsste.

I: Was ist dir beim Manipulieren von Objekten auf Distanz positiv oder negativ aufgefallen? Speziell bei dem Schneemanntask?

T1: Das an den Achsen entlang bewegen funktioniert soweit ganz gut, allerdings, was immer noch schwierig ist, trotz dieser Geschwindigkeits-Sache, ist, dass du immer noch relativ viel Arbeit hast Objekte über größere Distanzen zu bewegen. Das fand ich schwierig, oder verhältnismäßig aufwändig. In dieser Richtung finde ich zum Beispiel Go-Go einfacher als Technik.

I: Wie komplex ist deiner Meinung nach diese Technik zu erlernen? Würdest du sagen es ist eher etwas komplexes, oder etwas das schnell erlernbar ist?

T1: Ich glaube man kann es ganz gut erlernen, ich glaube aber, was man vielleicht machen sollte, ist ein Entweder-Oder anzubieten. Weil du im Moment viele Optionen für dieselbe Sache anbietest. Das macht es wesentlich komplexer als es eigentlich ist. Wenn du die normalen Sachen rausstreichen würdest, und das mit normaler Bedienung vergleichst hast du nicht mehr das Problem dass es zu viele Möglichkeiten gibt, was die Komplexität darstellt. Das ist kein Problem der eigentlichen Technik, sondern, dass du beide zulässt. Das finde ich schwieriger. Ich fände es geschickter deine Technik mit der regulären Interaktion zu vergleichen. Ich finde es eigentlich nicht komplex, das komplexe ist die Möglichkeit Sachen wie gewohnt zu machen, oder anders. Teilweise fand ich das Anders (mittels PROMT, Anm. d. Verf.) geschickter, aber es gab auch Sachen bei denen ich mich nicht darauf eingelassen habe, weil ich es anders gewohnt bin. Da finde ich eine weitere Option zu haben eher verwirrend. Aber ich glaube nicht, dass die Komplexität zu hoch ist.

I: Wie würdest du die Präzision einschätzen, welche du in dieser Applikation erreichen könntest?

T1: Das hängt auch davon ab. Das Grid-Snapping ist eine gute Sache, das funktioniert natürlich ganz gut. Bei dem Schneemann wird die Präzision gar nicht so hoch gewesen sein weil es auf Distanz war. Weil es relative traditionell auf Distanz war, zumindest so wie ich es genutzt habe. Ich glaube schon, dass es einigermaßen präzise geht. Was mir dazu im Gegenteil einfällt, ist zum Beispiel wenn ich beschleunige und dadurch eine größere Translation habe. Aber wenn du natürlich auch langsamer interagieren kannst und damit eine höhere Präzision erreichen kannst, kann das auch zuträglich sein. Schwer zu sagen, ich glaube man kann präzise damit arbeiten, das kostet

dann aber auch Zeit. Das ist auch wieder so ein Trade-Off.

I: Hattest du das Gefühl, dass du mit dem Ausrichten an der Achse schneller eine gewisse Achse manipulieren konntest und diese aber auch präziser manipulieren konntest?

T1: Ja, mit dem Hilfsmittel des Snappens auf jeden Fall. Also im zweiten Task ging das wesentlich zügiger. Ich müsste das aber auch mit der anderen Sache direkt vergleichen. Ich fand das ging ziemlich schnell und ziemlich intuitiv.

I: Gut, das waren die Fragen. Wenn du noch irgendetwas anmerken möchtest dann . . .

T1: Ich finde es gut! Es gefällt mir wirklich ganz gut. Ich kannte so etwas von der Idee her auch noch nicht. Was ich vom Vergleich her fair fände, und das würde dem Vergleich auch gut tun, wäre wenn man die Techniken separat miteinander vergleichen würde. Es ist eine Variante dem Benutzer [sic] alles zu erlauben, auch die Sachen zu mischen, was grundsätzlich ganz gut ist. Aber so wird die Komplexität der Geschichte höher. Das wäre ein Punkt den ich nicht schlecht fände.

I: Mein Gedanke war beides zu erlauben, weil es eine Testapplikation sein soll. Ich wollte meine Technik in einer Applikation unterbringen in der sie vielleicht später auch sein könnte. Ich bin mir ziemlich sicher, dass es stimmt, dass die Applikation dadurch sehr viel komplexer wird. Besonders wenn man (die Applikation, Anm. d. Verf.) nur sehr sporadisch verwendet, und nicht häufig. Ich habe mir dabei erhofft herauszufinden, ob Benutzer [sic] meine Technik verwenden, auch wenn sie die Wahl hätten etwas anderes zu verwenden. Um die beiden Möglichkeiten zu vergleichen ist es vielleicht nicht optimal.

T1: Was da natürlich auch schwierig ist, zu beachten welche User du hast. Wenn du jemanden hast der [sic] schon einmal mit so einem System gearbeitet hat, ist es eine ganz andere Sache. Wenn du beispielsweise einen Noob (Benutzer ohne Vorkenntnisse, Anm. d. Verf) davor setzt, der noch nie ein Headset (Head Mounted Display, Anm. d. Verf) auf hatte, und du erklärst ihm wie die manipulation funktioniert, dann kennt dieser keine andere Option. Da finde ich es fairer wenn man Benutzer [sic] ohne Vorgeschichte hat und diese das einfach ausprobieren lässt. Bei Virtual Reality Anwendungen, im Vergleich zu Desktop Anwendungen, bei denen du davon ausgehen kannst, dass jeder [sic] schon einmal eine Maus in der Hand gehabt hat, hast du den Vorteil, dass nur wenige einen Standard kennen. Das kann die Vorteile, aber auch Nachteile bringen. Was da zum Beispiel ein Nachteil ist, dass du in der Repräsentation von icons und widgets stark von 3D Modellierungswerkzeugen ausgehst, ist dass nicht jeder Benutzer [sic] diese kennt. Wenn du eine Studie hast mit Benutzern [sic] die weder das Eine noch das Andere kennen, würde ich fast vermuten, dass du auf andere Ergebnisse kommen wirst. Meine Annahme wäre, die Kritik mit der Komplexität würde rausfallen, weil du nur noch ein eingeschränkte Set an Optionen hast.

Damit könnte man auch einen fairen performance Vergleich zusammenzaubern, bei dem du feststellen könntest, vielleicht sind die Benutzer [sic] mit der einen Technik einfach schneller wenn sie beide Sachen erlernen müssen.

I: Das wäre auf jeden fall ein Schritt für eine größere Evaluierung. Ich glaube für eine Experteninterview [sic] bei dem ich einen kleinen Kreis von Menschen befrage und keine direkten Performance Vergleiche mache, hat es gut funktioniert, um grundsätzliche Probleme zu entdecken. Danke für das Interview!

A.2 Interview 2

I: Beginnen wir mit dem ersten Task. Beim ersten Task siehst du vor dir einen Tisch und du hast links neben dir ein paar Objekte die ein Geschirr darstellen. Deine Aufgabe in dem Task ist es, den Tisch zu decken, also einen Teller hinzulegen, das Glas hinzustellen. Du solltest darauf achten, dass die Objekte sich nicht überschneiden und dass der Teller und das Glas möglichst gerade auf dem Tisch stehen, sodass man dann gleich essen kann.

T2: Als erstes würde ich jetzt den Teller platzieren. Dazu habe ich den Teller jetzt mal ausgewählt. Dann würde ich schauen, dass ich den Teller in die richtige Richtung bringe. Ich hätte mit dem Auswahl-Button das Objekt verschoben, daran muss ich mich erst gewöhnen, ich habe jetzt den Grip-Button genommen. Das ist jetzt erledigt, das heißt ich habe mich hingekniet und das ausprobiert. Dann nehme ich als nächstes das Glas: genau das gleiche wie vorher, das heißt ich gehe hin und dekoriere es so, dass es dann auch wirklich auf dem Tisch steht, jetzt schaue ich es mir von oben noch mal kurz an und fertig.

I: Okay, sehr gut. Das war der erste Task. Dann gehen wir zum nächsten Task. Beim zweiten Task geht es darum, dass du eine Mauer baust mit einem Fenster darin. Du hast links neben dir ein Panel liegen, das kannst du verwenden. Es gibt auch ein paar Voraussetzungen: 1.) Die Mauer muss am Boden stehen. 2.) Die Mauer muss zwei Meter hoch sein.

T2: Also größer als ich.

I: Genau. Und 3.) in der Mauer muss ein Fenster sein und das sollte 50 mal 50 Zentimeter groß sein.

T2: Muss es genau die Größe haben oder ist das nur ein Richtwert?

I: Du solltest möglich nahe an das rankommen.

T2: Okay, das wird dann wahrscheinlich ein Meter mal ein Meter sein. Ist das Ding ein Meter groß?

I: Wenn du es auswählst und im Menü schaust, dann siehst du die Eigenschaften.

T2: Wo stehen die?

I: Jetzt ist es nicht mehr ausgewählt. In der Mitte siehst du die Position und die Skalierung.

T2: Das heißt, Position und Skalierung ist ein Meter mal ein Meter und fünf Zentimeter Dicke, oder?

I: Genau. Wo diese Wand steht ist egal, wichtig ist nur die Größe.

T2: Es sollte wirklich einfach eine durchgehende Wand sein? Das heißt, wir brauchen jetzt keine Burg sondern wirklich nur eine Wand auf einer Seite?

I: Genau.

T2: Passt, dann würde ich mal loslegen. Als erstes würde ich das Ding aufheben und ich ziehe mir das Ding her. Ich habe es mit dem Grip ausgewählt und ziehe es her, dass es so ungefähr passt. Jetzt würde ich mir den Parameter anschauen, weil der Rotationswert wird jetzt wohl nicht so passen. Jetzt gehe ich auf rotieren, die Achse interessiert mich.

I: Du kannst auch die Rotationsschritte einstellen, damit tust du dir vielleicht leichter, dass du eine gewisse Rotation bekommst.

T2: Kann man da rückgängig auch was machen?

I: Rückgängig geht nicht, nein.

T2: Ja, im Prinzip passt es eh nicht so schlecht. Außer es muss ganz genau sein?

I: Je genauer, desto besser.

T2: Okay, dann machen wir es noch mal so. Gut, dann gehen wir auch Transformation. Achso, wie ziehen wir die Mauer überhaupt auf?

I: Du kannst zum Beispiel auch kopieren.

T2: Passt, dann machen wir das so. Wir kopieren das ganze mal. Es muss ja kein durchgängiges Ding sein, weil es ist ja schon ein Meter mal ein Meter. Dann kopieren wir das ganze mal. Okay, jetzt ist das ein Meter mal ein Meter. Und das Fenster soll hier 50 Zentimeter breit sein, das heißt, 0,75. Okay, dann kopieren wir das einmal. Jetzt mache ich es ein wenig schmaler. Ich würde das gerne numerisch eingeben. Dann gehe ich wieder auf transformieren und duplizieren. So und jetzt brauche ich eigentlich noch einmal ein 0,5 und 50 Zentimeter soll es breit sein. Passt das hier so?

I: Ja, die Position ist dir frei überlassen.

T2: Passt, dann rein damit, Skalierung, Rotation, Snap. Ich hätte gerne das Menü auf dieser Seite.

I: Das Menü, wenn du es rufst, richtet sich nach deiner Kopfposition aus.

T2: Okay, super. Dann gehe ich auf skalieren und die Breite ist 0,5 und jetzt können wir es eigentlich neu positionieren.

I: Sehr gut, dann speichern wir das Meisterwerk ab und wir gehen zum nächsten und letzten Task. Beim dritten Task geht es hauptsächlich darum, dass du auf Distanz arbeitest. Das heißt, du hast diese Objekte vor dir und du kannst nicht hingehen, du musst sie quasi aus der Entfernung bearbeiten. Das Objekt ist ein riesiger Schneemann. Das heißt du hast die drei Kugeln, die stapelst du übereinander, um einen Schneemann zu bauen, dann gibst du ihm zwei Augen und eine Nase und ein paar Knöpfe. Wie genau er aussieht, ist dann wieder ein bisschen dir überlassen, kreative Freiheit.

T2: Gut, dann mache ich das mal. Dann wähle ich jetzt diese Kugel aus. Nachdem ich jetzt nicht ran kann, kann ich jetzt nicht coolerweise die Sachen aufheben aber das hat sicher einen bestimmten Grund, warum das so ist. Im Prinzip mache ich jetzt immer das Gleiche. Ich werde jetzt mit dieser (Achse, Anm. d. Verf.) ein bisschen rausfahren, sonst verdecke ich mir das Ganze. Dann fahre ich rüber, damit die Nase ungefähr genau ist. Dann rauf, rauf, rauf, und rotieren soll ich ihn auch noch. So, jetzt wird es lustig. Wir müssen die rotieren mit der Kurbel. Dann transformieren und rein mit der Nase. Perfekt, hier müssen die Augen rein. Kann ich es nicht duplizieren? [lacht]

I: Grundsätzlich ja.

T2: Das geht eigentlich leichter, dann mache ich das. So dann kopiere ich das und fertig.

I: Gut, dann speichern und das war es dann mit den Tasks und du kannst die Brille wieder abnehmen und ich würde dir noch ein paar Fragen stellen.

T2: Gerne.

I: Die erste Frage wäre: Inwieweit hast du das Ausrichten des Controllers an der Achse als nützlich empfunden, um separat die Achse zu selektieren?

T2: Also das heißt, dass durch dieses Drehen des Controllers in die richtige Achse auswählen? Ja sehr nützlich, ich hab das auch relativ schnell übernommen und es war für mich ziemlich intuitiv, obwohl ich es anders gewöhnt bin, weil ich es von den ganzen Konstruktionsprogrammen so nicht gewöhnt gewesen bin. Es war total nützlich, weil die Verbindung einfach da war. Es war für mich auch ein bisschen von der Distanz abhängig, weil ich ja verschiedene Szenarien hatte. Ich habe es bei dem Mauern-Task fast besser gefunden, weil es ein bisschen greifbarer war. Distanz war für mich dann schon ein wenig anstrengend die Sachen zu verschieben.

I: Warum würdest du sagen, war es für dich anstrengender auf Distanz?

L2: Weil ich mich zu erst immer konzentrieren musste, das richtige Ding auszuwählen sozusagen. Es war für mich einfach greifbarer, klarerweise, aufgrund des Abstandes. Wie gesagt, ich habe das sehr gut gefunden, ich habe auch nicht darüber nachgedacht, ich hab nur aber einer gewissen Zeit gemerkt, dass es in die Arme geht, das war aber erst zum Schluss. Aber ansonsten war es sehr intuitiv.

I: Was war für dich am kompliziertesten oder am wenigsten intuitiv beim Ausrichten? Ist dir irgendwas aufgefallen, was du am Anfang nicht gleich verstanden hast?

T2: Also verstanden schon, es ist ja ziemlich selbsterklärend. Das einzige mit dem ich öfters Fehler gemacht habe, war, dass ich dieses Auswahl-Werkzeug auch gleichzeitig als Manipulationsaktion gesehen habe beim Achsen verschieben. Ich habe zuerst nicht das Anfassen gedrückt, sondern sondern das Auswahl-Ding und dann habe ich mich gewundert, warum nichts gegangen ist. Aber sonst, war die Aktion an sich eigentlich ziemlich klar. Vielleicht war es deswegen ein bisschen mühsam, von einer größeren Distanz

Sachen zu manipulieren, weil die Strecken länger waren. Aber sonst, an sich war es gut nachvollziehbar.

I: Ist dir die skalierte Bewegung beim Translieren positiv aufgefallen oder gar nicht aufgefallen?

T2: Sie ist mir gar nicht aufgefallen. Wenn du es mir vorher nicht gesagt hättest, wäre sie mir gar nicht aufgefallen.

I: Was ist deine Meinung zur Kurbel-Mechanik beim Rotieren? War das verständlich?

T2: Wie gesagt, es war für mich nicht intuitiv. Ich wollte immer zum Rotieren anfangen, weil ich es bei den anderen Aktionen auch so war und du hast mir dann halt immer gesagt, schau bitte auf dieses Rotierwerkzeug, dann bin ich wieder reingekommen. Ich war mir auch bei manchen Achsen teilweise nicht sicher, wie ich es drehen muss, ob ich es so drehen muss, oder ob ich es fester drehen muss und dann muss ich eigentlich aus diesem Threshold raus; das war für mich dann auch von der Sicher her nicht so klar, bin ich jetzt schon draußen? Wenn ich es jetzt vergleiche mit den anderen Methoden, war das jetzt so die, bei der ich am meisten Schwierigkeiten hatte. Es hat mir ein bisschen zu lange gedauert, bis das es angesprungen ist, es war ein bisschen schwerfällig.

I: Okay. War das haptische Feedback beim Manipulieren nützlich, um zu wissen, wann du etwas manipuliert hast? Oder war es eher unverständlich, wo das Feedback hergekommen ist?

T2: Damit meinst du die Vibration?

I: Genau.

T2: Ich muss zugeben, darauf habe ich gar nicht geachtet. Das ist mir gar nicht aufgefallen. Ich habe es einmal wahrgenommen, aber eigentlich war es für mich gar nicht so stark. Ich habe mich eigentlich viel mehr auf das visuelle konzentriert, also ich habe es gar nicht wahrgenommen. Vor allem war der erste Task für mich einfach schon so ganz anders, das war für mich der direkteste Task, das war für mich wirklich so dieses Greifen, da war für mich schon ein "haptisches Feedback" da, obwohl eigentlich gar keines da war, das hat sich für mich schon so gut angegriffen.

I: Vielleicht noch mal kurz zur Erklärung. Das haptische Feedback ist dann aufgetreten, wenn du eine einzelne Achse selektiert hast, wenn du sie frei bewegt hast, dann hat es keines gegeben, wenn du eine Achse selektiert hast, dann schon.

T2: Ja, jetzt da du es sagst, fällt's mir ein. Aber mir ist es gar nicht aufgefallen, ich habe einfach so fest zugegriffen.

I: Welche Probleme hattest du beim Verwenden von dem Ray-Cast Selection Tool?

T2: Prinzipiell eigentlich keines. Nur bei der Schneeman-Task, dass ich halt treffe, aber die Ziele waren auch kleiner und schwerer zu treffen. Was die Objekte betrifft. Was ich schwierig gefunden habe, ist, beim Menü selber, weil das ja ein bisschen anders funktioniert, das ich die Sachen auswähle. Da

wollte ich zuerst nämlich mit diesem Ray-Cast den Knopf auswählen und dann auch wieder drehen, was ja eigentlich ein Blödsinn ist in dem Fall. Aber sonst, das Auswahl-Werkzeug an sich funktioniert sehr gut, bei Distanzen halt dann nicht mehr so, aber auch noch in Ordnung.

I: Okay. Hast du irgendwie bemerkt, dass du Probleme hattest, Sachen auszuwählen, wenn du wolltest oder hat das für dich immer funktioniert?

T2: Prinzipiell schon. Nur halt diese Distanzen, wenn etwas kleiner war, habe ich nicht so gut getroffen, aber sonst eigentlich nicht. Was mir auch manchmal nicht klar war, aber das hat weniger mit dem Auswahl-Werkzeug an sich zu tun, war, ob etwas ausgewählt ist, oder nicht. Die Farben waren relativ ähnlich, das eine war blau und das andere türkis, wegen dem Highlighting, da habe ich vielleicht nicht so darauf geachtet. Da war ich mir nicht klar, was habe ich jetzt aktiv ausgewählt, aber das hat mit dem Tool nichts zu tun.

I: Wie sehr hat dich das Wechseln zwischen den Manipulationsmethoden unterbrochen, von dem, was du gerade gemacht hast? Wie sehr hat es dich abgelenkt, wenn du zwischen translieren und rotieren und skalieren umschalten musstest?

T2: Prinzipiell nicht so wirklich. Ich musste halt immer schauen, wo was ist, aber das ist eine Übungsfrage, dass ich weiß, dass links das translieren, oben rotieren und rechts skalieren ist. Ich musste dann halt immer schauen, ist das wirklich so. Aber eigentlich nicht so wirklich, nein. Was mir dann in der Anfangsszene aufgefallen ist, weil die Objekte relativ klein und handhabbar waren, das war das gute, weil alle Modi auf einmal dabei waren. Aber ich habe versucht immer ein bisschen zu mixen. Bei der Wand-Task ist es natürlich nicht gegangen, weil da Präzision gefordert war und da musste ich dann schon immer schauen zwischen den Modi. Ich glaube, das ist eine Übungsfrage, ich glaube, ich hätte noch ein bisschen warm werden müssen damit, dann wäre ich schon reingekommen.

I: Okay. Was hat für dich gut funktioniert, entfernte Objekte gut zu manipulieren? Was ist dir da positiv aufgefallen, wenn du Objekte manipuliert hast, ohne, dass die in deiner Reichweite waren?

L2: Es war am Anfang ein bisschen ungewohnt, ich wollte am Anfang hingehen. Aber was mir sehr gut gefallen hat, und das hat auch immer funktioniert, war dass man die Achse ausrichten konnte. Das war immer ziemlich genau. Das bewegen entlang dieser fixierten Achsen, ohne dass ich die Achsen ausgewählt habe, hat mir gut gefallen.

I: Okay. Jetzt generell zur Applikation: Wie komplex hast du die Interaktionen gefunden? Hattest du das Gefühl, dass es eher etwas ist, was man länger erlernen muss oder fandest du es relativ straightforward, leicht zu erlernen?

L2: Die Komplexität ist ansich ist okay. Es sind halt einfach Abläufe bei denen man Zeit braucht, aber das hat nichts mit der Komplexität zu tun, sondern dass man sich manche Sachen einfach lernen muss, weil man damit

nicht so viel Erfahrung hat. Auch wie ich vorhin schon erwähnt habe, diese Button-Belegung, das muss erst einfach in Fleisch und Blut übergehen. Aber das war keine komplexe Zeichen-Kombination die man erlernen muss, da muss man einfach lernen nicht mehr hinzuschauen. Wobei man es eh am Objekt sieht, in welchen Modus man gerade drinnen ist.

I: Also eher weniger komplex.

T2: Genau. Und was ich sehr gut gefunden habe, ist, dass es diese verschiedenen Modi auch gibt. Dass man es sich aussuchen kann und dann auch flexibler ist. Dass ich Achsen auswählen kann zum Beispiel.

I: Wie präzise hattest du das Gefühl, dass man in der Applikation sein kann?

T2: Wie du gesagt hast, man kann sicher diese Positionen, Rotationswerte und so, die sehe ich zwar, aber es ist eher eine Daumen mal Pi Geschichte, natürlich könnte man es auch verwenden, aber dann wäre ich gleich von vornherein ganz anders rangegangen, an die Geschichte. Also dass ich nicht mir irgendwelchen Kommawerten arbeite, weil das kann man dann eh einstellen, was du genau machst. Ich wollt zuerst nämlich auch das Ganze von vorne aufziehen, machen, aber das ist leider nicht gegangen. Ich glaube, man braucht fast eine numerische Eingabe, dass man genau ist. Ich glaube mit hinschieben und snappen und wechseln und so, kann es ein bisschen mühsam werden. Ich meine, bei so Sachen wie Wände einziehen das geht, wenn es relativ einfach noch bleibt. Aber Präzisionsarbeit ohne ein zusätzliches Hilfsmittel glaube ich eher weniger. Es geht einfach schnell ein Mock-up zu machen, Ideen zu skizzieren, dafür finde ich es gut. Für Präzisionsarbeiten, mit dem was jetzt zur Verfügung steht, sehe ich jetzt momentan nicht so das Potential in der jetzigen Form.

I: Okay. Wenn du möchtest, wäre jetzt noch Zeit für ein freies Feedback, wenn dir noch etwas einfällt.

T2: Prinzipiell hat mir die Sache voll gut gefallen. Ich finde es voll witzig, man bekommt einfach, eben wenn man so 3D-Programme kennt vorm Desktop sitzend, fühlen sich die Sachen doch ganz anders an. Du hast einfach viel mehr Gefühl, dass du in dieser Szene drinnen bist, du arbeitest an einem Objekt und nicht an einem abstrakten Ding am Bildschirm. Was mir sehr gut gefällt, ist dieser frei-Fahr-Modus, wo du Sachen positionieren kannst, das gefällt mir gut und auch trotzdem diese Möglichkeit auf Achsen zu arbeiten, dass du jetzt nicht komplett frei unterwegs bist, das fühlt sich gut an. Die Auswahl finde ich auch gut. Ich finde das Projekt voll interessant, weil man die Potentiale auslotet. Ich finde teilweise die Beobachtung ziemlich interessant, weil trotzdem versucht wird die Icon-Sprache vom Desktop versucht wird zu übersetzen. Manche Sachen funktionieren gut und bei manchen Sachen denke ich mir, dass man sie vielleicht gar nicht mehr so braucht, weil sich das ganze einfach komplett anders anfühlt, das ist irgendwie nicht das Gleiche wie am Desktop. Also spannendes Projekt, cool.

I: Danke!

A.3 Interview 3

I: Ich starte jetzt den ersten Task. Bei dem siehst du rechts von dir einen grauen Tisch stehen und die Aufgabe ist es, diesen grauen Tisch zu decken. Du hast links neben dir drei Objekte, die sollen Glas, Teller und Früchte darstellen und die sollst du jetzt auf den Tisch platzieren. Du solltest dabei darauf achten, dass du die Objekte möglichst aufliegend platzierst, so gut wie möglich und dass es möglichst wenige Überschneidungen von Objekten gibt.

T3: Mit der Zweiteilung zwischen Trigger-Button und den äußeren, da drücke ich immer irgendwie das falsche.

I: Das ist momentan auch noch eines der größeren Probleme.

T3: Das sieht ganz okay aus. So ungefähr, ist das so präzise genug?

I: So präzise wie du es schaffst, das ist die Herausforderung. Wenn du sagst, dass es fertig ist, dann ist es fertig.

T3: Okay, dann mache ich noch ein bisschen. So, Tisch ist gedeckt. Ich hätte mir das auch ein bisschen leichter machen können, wenn ich die Rotation eingestellt hätte. Das mache ich dann beim nächsten Mal.

I: Gut, dann gehen wir zum nächsten Task, da wird die Szene wieder gewechselt. Bei diesem Task siehst du gleich neben dir dieses Panel, das ist ein Wand-Panel.

T3: Moment, ich habe hier ein Fuß-Panel.

I: Ja, genau, das Panel liegt am Boden.

T3: Okay.

I: Auf diesem Panel sollst du eine Wand vor dir bauen. Diese Wand soll aber ein paar bestimmte Eigenschaften haben: sie soll möglichst genau zwei Meter groß sein und es soll ein Fenster in dieser Wand sein, das 50 Zentimeter in beiden Achsen groß sein soll. Die Position der Wand ist dir überlassen und wie du die Wand aufbaust ist dir überlassen. Du kannst dieses Panel kopieren im Menü oder eigene Objekte aufziehen, wie du willst, aber wichtig ist, dass die Wand zwei Meter groß ist und ein 50 Zentimeter großes Fenster darin ist.

T3: Einen Moment. Das Panel soll auf die Wand montiert werden?

I: Nein, das Panel ist Teil der Wand. Das ist nur ein Anfangsteil mit dem du arbeiten kannst.

T3: Alles klar. Also zwei Meter große Wand, zwei Meter hoch, 50 Zentimeter Fenster. Okay, dann probiere ich das mal. Ich versuche gerade das Ding zu rotieren, ich hatte jetzt gerade Schwierigkeiten die Rotation auszulösen. Es ist eine sehr geometrische Aufgabe. Jetzt machen wir das wie vorher. Ich mache jetzt mal zehn Zentimeter Schritte und ich mache jetzt 45 Grad. Jetzt schnappe ich mir das, moment wie war das nochmal: freie Rotation wenn ich gar nichts auswähle?

I: Genau.

T3: Das ist aber jetzt gar nicht so verkehrt im Moment. So, dann muss

ich eine Translation machen. Ich weiß jetzt nicht wie weit ich ...

I: Es wird ein Sicherheitsgitter angezeigt, sobald du zu weit rauskommst.

T3: Okay, ich wollte jetzt nämlich genau ein Sicherheitsgitter anbringen. Aber das war jetzt vielleicht gar nicht notwendig. Also das muss gar nicht parallel zum Sicherheitsgitter sein?

I: Nein.

T3: Ah, okay, dann habe ich die Aufgabe jetzt ein wenig überinterpretiert. So, ja gut, dann lasse ich das jetzt erst mal. So, jetzt zwei Meter. Jetzt muss ich erst mal hier Translation machen, ich kann das auch aus der Entfernung wenn ich das richtig mache, Objekt kopieren. Aha, jetzt wird es also größer.

I: Du bist jetzt im Skaliermodus.

T3: Mhm. Ja, puh, ich überlege jetzt gerade - jetzt machen wir erstmal eine Translation. Jetzt stelle ich das Ding auf den Boden und jetzt muss ich das Ding hoch machen.

I: Wenn du die Achse einzeln selektieren willst, dann siehst du anhand der Farbe der Achse, wenn die opak wird, dann ist sie selektiert mit der Ausrichtung vom Controller.

T3: Ich habe jetzt gerade noch ein Problem die Größe abzuschätzen.

I: Am Menü ist noch die Skalierung angezeigt, die ist in Metern. Also das Objekt ist jetzt 1,80 Meter hoch und 2,50 Meter breit.

T3: Aha, das ist sehr hilfreich, das brauche ich.

I: Das Menü wird anhand deiner Kopfposition platziert, wenn du es rufst.

T3: Ah okay gut, ich wollte das jetzt quasi über die Mauer legen, dass ich draufgucken kann. So, alles klar. Dann mache ich jetzt hier mal weiter. Achsen auswählen ist für mich manchmal noch ein bisschen [schwer, Anm. d. Verf.]. So, da haben wir jetzt zwei Meter. Jetzt hier zwei Meter. Tiefe war nicht definiert, oder?

I: Nein, die Tiefe ist dir überlassen.

T3: Aha, aber das ich jetzt gar nicht so leicht ausschneiden fällt mir gerade auf [lacht].

I: Das Ausschneide-Tool fehlt leider.

T3: Ah! Okay gut, dann muss ich jetzt erst mal so ein Ding bauen. Oh gott, da brauche ich ja ewig. Jetzt machen wir den mal so, dann dass, dann wollen wir das kopieren.

I: Du wählst gerade die y-Achse, oder?

T3: Ja.

I: Dann musst du den Controller nach oben zeigen.

T3: Am liebsten würde ich so machen.

I: Das müsste auch funktionieren, wenn du den Würfel hier berührst in der Mitte kannst du es auch händisch machen.

T3: Ah okay, ich glaube das mit dem Würfel habe ich noch nicht ganz geschnallt. Das ist dann schon ein bisschen einfacher. Wie hoch soll das Fenster sein?

I: 50 mal 50.

T3: Puh, das ist ja richtige Arbeit! [lacht]

I: Eine Mauer zu bauen ist nicht einfach. [lacht]

T3: Ja, ja ich merke schon. So, jetzt nehmen wir den hier, dann probieren wir das noch mal. Das ist so lustig was man für Quatsch macht: ich versuche die ganze Zeit meiner Mauer nicht entgegen zu rennen [lacht]. So, jetzt machen wir in der Mitte den hier, den mache ich jetzt kleiner.

I: Es hat schon funktioniert, es ist nur bei der Skalierung jetzt auch zehn Zentimeter-Schritt drinnen, das richtet sich nach der Grip-Size aus, das du eingestellt hast.

T3: Ach so, das muss ich dann jetzt anders machen. Ich brauch dieses Ding hier jetzt. Muss die Mauer gleich dick sein?

I: Nein, das war nicht die Aufgabe.

T3: Aber Moment, das stimmt doch irgendwie nicht, das ist doch jetzt nicht da. Das ist jetzt ja nicht 50 wenn das andere zwei Meter ist, was hab ich denn jetzt gemacht?

I: Die Breite ist jetzt 50, oder? Die Tiefe ist jetzt 50.

T3: Na gut, aber macht das Sinn wenn ich das jetzt umdrehe? Einen Moment. Aber ist das ist nicht quadratisch. Warum ist das nicht quadratisch? Meine Güte!

I: Du musst es wieder auswählen.

T3: So, das ist nicht quadratisch in eine Richtung.

I: Die x-Achse.

T3: Das ist die hier, oder?

I: Genau. Vielleicht ist das noch nicht ganz klar. Der Griff von dem Controller ist parallel zur diese x-Achse.

T3: Ah, okay, ich glaube, das habe ich bisher noch nicht geschnallt. So, hui, so dann verschieben noch einmal. Mist! Die Fuge stimmt nicht ganz. Aber okay, jetzt machen wir erst mal fertig hier. So, copy, copy.

I: Es ist schon selektiert, es ist nur verschoben.

T3: Ah, okay, verschieben, so runter, gut. Du merkst, mit 3D und VR habe ich sonst nicht so viel zu tun.

I: Das ist absolut in Ordnung.

T3: Naja, präzise ist es nicht ganz. [lacht]

I: Es ist auf jeden Fall die massivste Mauer bis jetzt. [lacht]

T3: Ja, das denke ich auch. Das ist aber auch kein Fenster mit 50 mal 50 Zentimeter, wieso ist das jetzt? Das hier ist 50, aber das ist ja gar nicht 50. Ich muss das jetzt hier noch größer machen, oder? Das stimmt auch nicht, warum nicht?

I: Ein Teil davon ist jetzt im Boden versunken, weil das natürlich in beide Richtungen skaliert.

T3: Also das was wirklich helfen würde, ist, wenn die Physik da wäre und die Sachen einfach auf dem Boden stehen bleiben würden. [lacht]

T3: Ah, okay, das würde mir etwas helfen. Das müsste jetzt so ungefähr

50 mal 50 sein. Das war eine schwere Nummer.

I: Bist du zufrieden damit?

T3: Ja, ich bin zufrieden. [lacht]

I: Gut, dann starten wir mit dem letzten Task weiter. Der dritte Task ist: Baue einen riesigen Schneemann. Vor dir sind diese Objekte und die sind bewusst außerhalb deiner Reichweite. Deine Aufgabe ist einfach, diese Kugeln zu einem Schneemann zusammen zu stapeln und den dann mit Augen und einer Nase zu verzieren.

T3: Okay. Das was schon relativ gut funktioniert, dass du diese beschleunigte Bewegung hast, das hilft mir jetzt doch ziemlich, dass ich die Sachen jetzt zusammenbekommen habe. Jetzt muss ich aber wieder ins Menü, damit ich einigermaßen orthogonal aneinander ausgerichtete Nase hinbekomme. Ich würde ihn ja gerne von der Seite noch sehen können, das wäre jetzt schon auch noch hilfreich. Okay und jetzt die Augen und die Knöpfe noch. Das sieht gar nicht so verkehrt aus glaube ich. Abschließend beurteilen könnte man es echt nur mit einer Rundum-Ansicht, aber soweit ich das beurteilen kann, sieht das gar nicht mal so schlecht aus. Das ging wirklich super easy. Diese Bewegung und dass man damit auch große Distanzen überwinden kann, ich bin jetzt nicht so wahnsinnig schnell, weil ich ein bisschen zögere, den Controller in die richtige Richtung zu werfen und weil man sich ja nicht ganz so sicher ist wo man gerade ist und was man zerschmettern könnte. [lacht] Ich habe schon mal einen VR-Controller an einer Wand zerhauen, also so gesehen, das ist so ein Überlebensinstinkt, dass man dann die Bewegung nicht ganz so krass schnell macht. So, hier ist natürlich das Treffen schon ein bisschen anspruchsvoller. Das ist interessant, ich glaube es wäre nicht schlecht, wenn ich das Feedback früher bekommen würde. Also es ist so, immer erst wenn ich beim Button 'release' ... also hier jetzt gerade gar nicht mehr ... also irgendwie die gleichen Targets sind ein bisschen schwer zu treffen unter anderem auch weil ich nicht direkt visuelles Feedback darauf bekommen bei mouse-over, ob ich das jetzt getroffen habe oder nicht, das hat mich jetzt ein paar Mal irritiert. Das wäre vielleicht gar nicht so schlecht, wenn man da eine Voransicht hätte, damit man weiß, inwieweit ist es, ... ja, den finde ich schön, sehr beeindruckend.

I: Sehr gut. Das war auch schon der letzte Task.

T3: Darf ich damit auch jetzt noch ein bisschen rumspielen oder brauchst du das Modell zur Auswertung?

I: Das Modell ist gespeichert, du kannst gerne damit rumspielen.

T3: Okay, ich will den Kopf abnehmen. [lacht] Sieht sehr schön aus!

I: Gut, sobald du fertig bist, kannst du das Headset abnehmen.

T3: Wie lange habe ich jetzt dafür gebraucht?

I: Ich weiß jetzt nicht mehr wie lange die Einführung gedauert hat, aber es waren ungefähr 25 Minuten.

T3: Aha, das ist schon ein interessanter Punkt, wie sehr das einen auf die Dauer schlaucht oder nicht schlaucht.

I: Ja, das stimmt. Gut, dann hätte ich jetzt noch ein paar Fragen. Die erste Frage ist: Wie nützlich hast du das Ausrichten des Controllers an den Achsen empfunden, um die einzelnen Achsen zu selektieren?

T3: Ja, das habe ich ehrlich gesagt, wie du ja nach deinem Kommentar festgestellt hast, am Anfang nicht ganz so geschnallt. Ich hab eher relativ den Controller bewegt, bis ich dann das Feedback hatte, jetzt ist die Achse ausgewählt und nachdem ich dann verstanden hatte, ach so, wenn ich absolut den Griff des Controllers in die Richtung der Achse bringe, dann wars doch effizienter, deswegen habe ich mich da ein bisschen trottelig angestellt. Also ich glaube, das ist eine sehr nützliche Idee, man muss bloß kapieren, dass es um eine absolute Richtungseingabe geht und nicht um so lange relativ bewegen bis man die richtige Selektion hat. Das war mir einfach nicht so klar, deswegen habe ich mich da so schwer getan.

I: Was würdest du sagen, war das komplizierteste daran, diese Achsen zu selektieren mit der Ausrichtung des Controllers?

T3: Ja, also das komplizierte wenn man es verstanden hat, war es dann nicht mehr kompliziert. Ich habe halt relativ unkoordinierte Bewegungen gemacht bis ich das visuelle Feedback hatte, jetzt ist die Achse ausgewählt. So gesehen habe ich Roulette gespielt und irgendwas gemacht in der Hoffnung, dass ich jetzt endlich mal die Achse richtig auswähle. Das war eigentlich das komplizierteste daran. Als ich dann das Prinzip verstanden hatte, war es dann deutlich einfacher, hoffe ich zumindest. Zumindest weiß ich jetzt, wie es funktioniert.

I: Wie hat dich die skalierte Bewegung der Objekte beeinflusst? Also die skalierte Translation mit schneller Handbewegung.

T3: Das fand ich, wie ich vorher schon erwähnt habe, richtig praktisch, weil eben dadurch recht große Distanzen durch relative kleine Bewegungen überbrücken konnte. Ich hätte mir wahrscheinlich manchmal sogar ein bisschen mehr gaine gewünscht, sozusagen, zwischen motor space und virtuellem space, weil ich so dazu genötigt war, relativ schnelle Bewegungen zu machen, die mir persönlich ein bisschen unangenehm waren, weil ich den Raum hier nicht kenne und ich dann Angst hatte, ich könnte irgendwie in der physischen Welt mit etwas kollidieren. Aber prinzipiell finde ich das ein sehr cooles Konzept, ich würde vielleicht insgesamt sogar diese Gain-Rate noch ein bisschen höher drehen, damit noch kleinere Bewegungen noch größere Distanzen zurücklegen. Wobei es dann natürlich auf der anderen Seite wieder schwierig wird, wenn man etwas sehr präzises macht, aber wenn es so eine Kurve ist, dann dürfte das eigentlich gehen.

I: Eine Lösung für diese Problem, die ich mir schon überlegt habe seit der Evaluierung, ist die Distanz des Objektes mit einzubeziehen, wenn das Objekt weit weg ist, die ganze Beschleunigung noch mehr zu erhöhen.

T3: Ja, dass kann gut sein, weil vielleicht war das der Grund, warum mir bei den Kugeln beim Schneemann, da hatte ich das Gefühl, das geht mir völlig leicht aus der Hand, während es mir bei dem Mauer-Beispiel, ja gut

beim Schneemann waren die auch einfacher auf einer gewissen Distanz und da hat es sich sehr gut angefühlt und ich wüsste jetzt nicht, wenn die jetzt deutlich näher gewesen wären, ob ich gleich begeistert gewesen wäre. Das weiß ich jetzt nicht so genau. Wie gesagt, gain, das ist sicher eine interessante Stellschraube. Ich weiß auch nicht genau, wie du das bereits machst, da habe ich jetzt nicht genau darauf geachtet, wobei bei der Mausbeschleunigung ist es ja auch so, also beim Mauspfeil, da hat man ja diesen controll-space und zwischen dem display-space so ein gain und da haben die ja auch so eine Funktion, dass die gerade am Anfang bei sehr kleinen Bewegungen noch einen sehr kleinen gain haben und dann geht es nicht aber linear, sondern in so einer, ich weiß es nicht genau, wie es bei dir aussah.

I: Bei mir ist es nicht linear, es wird auch runterskaliert die Bewegung, also wenn du dich langsam bewegst, dann bewegst du das Objekt noch langsamer.

T3: Ja, genau. Aber das ist glaube ich eine gute Idee, linear ist hier glaube ich zu einfach.

I: Wie hast du diese Kurbelmechanik empfunden? War das für dich eine Metapher, die du leicht verstanden hast?

T3: Ich hab mich da immer ein bisschen schwer getan. Wenn ich mich darauf konzentriert habe, dann habe ichs schon verstanden. Ich fand es ja eigentlich eine coole Sache, weil dadurch, dass du den Radius erhöhst, kannst du die Präzision deiner Rotation erhöhen, also im Prinzip eigentlich echt gut. Ich habe mich bloß irgendwie dämlich damit angestellt. [lacht] Warum das jetzt so war, es hat halt sowas ein bisschen von einem Modus und da muss man halt wissen, dass man in dem Modus ist. Und ich habe glaube ich, das war vielleicht manchmal das Problem, ich glaube ich habe oft auf das Objekt geguckt, aber das Feedback, ob ich in dem Modus bin, oder ob die Kurbel gerade eigentlich an ist, war eigentlich nicht am Objekt, sondern eher am Controller und deswegen hatte ich glaub ich das Problem, dass ich, visuell mir manchmal nicht so darüber bewusst war, wie weit bin ich jetzt im Radius, also wo bin ich sozusagen mit der Kurbel. Hätte ich die Kurbel quasi noch mal in meinem Sichtfeld gehabt in irgendeiner Form, oder irgendwie Feedback dazu, wäre es mir glaube ich leichter gefallen.

I: Okay. Ich habe jetzt nicht extra darauf hingewiesen, es hat ein haptisches Feedback gegeben, das wirst du wahrscheinlich gemerkt haben.

T3: Ja.

I: Die Frage ist, wie befriedigend war dieses Feedback, um heraus zu finden, was man gerade bewegt oder wie sehr man gerade ein Objekt bewegt?

T3: Also wenn ich es recht erinnere, war es glaube ich immer so, dass, sobald ich diese Mittelfinger-Tasten bewegt habe, die Grip-Buttons hatte und damit eben so eine Operation gemacht hatte, wie Translation, Skalieren, Rotieren, dass währenddessen ich mit jedem Schritt auch ein Feedback bekommen habe und das fand ich insofern schon echt gut, weil es eben da unterscheidet von dem Modus wo du jetzt einfach nur den Controller be-

wegst ohne dass ein Objekt daran hängt, also das ist dann quasi so, als ob du irgendwas physisches tust und du bekommst dann quasi so physisches Feedback, das finde ich eine sehr gute Idee. Das fand ich gut, das fand ich hilfreich. Das ist nämlich genau das was ich vorher meinte, so ein Modus, du musst immer wissen, ob du in einem Modus bist in irgendeiner Art und Weise und das kommuniziert eben gleich auch noch bei jeder Bewegung, dass du im Modus bist und du brauchst gar nicht zwangsläufig Feedback, um zu wissen, okay, jetzt passiert hier gerade was mit dem Objekt.

I: Okay, sehr gut. Welche Probleme hattest du beim Selektieren vom Objekt mit diesem Ray-Cast?

T3: Ja, also das hat man ja am Schluss relativ stark, wenn es kleine Objekte auf eine weite Distanz sind, hatte ich das Gefühl, dass ich gar nicht so schlecht war beim Zielen. Aber ich habe dann, während ich noch den Daumen und Zeigefinger gedrückt hatte, noch kein Feedback, ob ich getroffen habe und bin deswegen die ganze Zeit davon ausgegangen, ich hab es jetzt noch nicht erwischt. Also wenn das quasi so wäre, dass man immer gleich ein Highlight sieht, welches Objekt der Ray-Cast sich jetzt auswählen würde, sobald ich den Finger loslasse, also quasi schon während dem hovern oder dem "mouse-over", wäre das für mich glaube ich einfacher gewesen. So musste ich zuerst immer den Finger loslassen, um dann festzustellen, ob ich getroffen habe und wenn ich nicht getroffen habe, musste ich den Finger dann wieder dran nehmen und dadurch verziehe ich ja schon wieder ein bisschen.

I: Okay. Und ansonsten hattest du das Gefühl, dass es gut funktioniert, Objekte damit zu selektieren?

T3: Ja, ja, das hat auf jeden Fall funktioniert. Wo du gerade von selektieren redest: Mit dem Ray-Cast hat das sehr gut funktioniert. Eigentlich, so das was mir am natürlichsten gewesen wäre, wäre das zu benützen mit dem Würfel und dem Objekt drinnen, aber da hatte ich oft das Problem, dass die zu weit weg sind. Das hätte ich eigentlich noch viel mehr benutzt, hätte ich nicht das Gefühl immer gehabt, ich stehe zu weit weg, weil das nämlich irgendwie so das absolut natürlichste wäre, einfach das Objekt in die Hand zunehmen, das ist so das Instinktivste. Aber das habe ich nicht so oft gemacht, wie ich eigentlich gedacht hätte. Warum nicht auch, weil ich immer ganz gut reingekommen bin.

I: Okay. Was würdest du sagen, war das größte Problem für dich beim Selektieren allgemein?

T3: Das hat eigentlich funktioniert. Also das Selektieren von Objekten war eigentlich nicht schwierig. Was doch ein bisschen anstrengend ist, weil man auf vieles achten muss, ist nicht nur das Objekt zu selektieren, sondern dann auch zu selektieren, okay, mit welchem Werkzeug bearbeite ich es jetzt und in welchem Modus bin ich jetzt und wie schalte ich jetzt um mit den anderen. Das hat jetzt wieder nichts mit dem Selektieren zu tun, sogesehen beantworte ich gerade deine Frage nicht [lacht], aber ich merke es jetzt trotzdem an. Ich fand es relativ anstrengend, zwischen den Werkzeugen so-

zusagen, zwischen den drei Operationen hin- und her zu wechseln und mich immer daran erinnern, wie wechsele ich jetzt zwischen denen, okay, da muss ich den Button nehmen und dann zu gucken, bin ich jetzt auch wirklich in dem Modus und wie mache ich jetzt die Manipulation. Also mit etwas mehr Übung, kommt das vielleicht viel flüssiger. Aber wie du bei mir gesehen hast, musste ich jedes Mal immer noch ziemlich viel nachdenken, ich musste mich ziemlich konzentrieren. Und das hatte mich eigentlich überrascht, weil eigentlich glaube ich, das konnte mir leichter von der Hand gehen aber ich würde da definitiv noch mehr Übung brauchen.

I: Ja, da ist auf jeden Fall noch ein bisschen Einübungszeit nötig für das Tool.

T3: Aber ich kann mir schon vorstellen, dass man dann sehr flüssig wird mit der Zeit.

I: Gut, du hast jetzt eigentlich gleich die nächste Frage beantwortet, nämlich wie sehr dich dieses Wechseln zwischen den Tools abgelenkt hat.

T3: Ja, also das fand ich erstaunlich schwierig, da musste ich mich doch schon ziemlich konzentrieren. Ich habe eigentlich aus der Darstellung vom Objekt, ich habe da zwar gesehen, o.k. das ist ausgewählt und das ist ausgewählt, aber ob das jetzt Rotation oder Translation oder Skalierung gerade ist, habe ich eigentlich eh immer nachgeguckt auf dem Controller, obwohl ich es eh eigentlich wahrscheinlich schon am Objekt hätte sehen können in der Art und Weise wie jetzt die Pfeile oder die Kurbel oder so. Aber ich habe mich einfach viel mehr auf diese Buttons auf dem Controller konzentriert, was dann dazu geführt hat, dass ich natürlich immer woanders hingeguckt habe. Das ist schon mal Belastung Nummer 1. Dann musst du erstmal identifizieren, welcher von denen ist aktiv, dann sind die Icons natürlich nicht besonders groß, ich habe da öfters mal Translation mit Skalierung glaube ich verwechselt, weil die Icons nicht so gut unterscheidbar sind mit der Auflösung die man hat. Und dann habe ich dann halt ziemlich lange gebraucht, okay, welches ist aktiv, welches sollte aktiv sein, habe dann ewig lang auf den Controller geguckt, dann war ich so weit und hatte dann Translation ausgewählt und dann okay, um welches Objekt ging es eigentlich noch? Dann musste ich erst wieder gucken.

I: Also eine extreme Ablenkung eigentlich.

T3: Ja, das hat mich schon abgelenkt. Was aber auch eben daran lag, dass ich die Information, die ich direkt im Sichtfeld habe, richtig verarbeitet habe, sondern immer auf den Controller geguckt habe, um zu schauen, welcher Modus gerade aktiv ist.

I: Okay. Würdest du sagen, dass die Indikatoren nicht eindeutig genug waren die auf dem Objekt drauf waren?

T3: Ja, letztendlich muss ich dann wohl sagen ja, obwohl ich jetzt nicht so genau was was man da jetzt viel besser machen könnte, aber offensichtlich war es für mich viel dominanter auf den Controller zu gucken und viel eindeutiger, als das was auf dem Objekt selber passiert, was eigentlich in-

teressant ist. Also wenn man das vielleicht kombinieren würde, also nehmen wir mal an, da würde jetzt so eine Icon-Darstellung irgendwie noch zusätzlich am Objekt sein, die so eindeutig ist wie am Controller, dann wäre das vielleicht nicht notwendig gewesen. Aber so habe ich mich unheimlich stark am Controller orientiert und an dem dort leuchtenden Button, warum auch immer.

I: Okay. Was hat für dich gut funktioniert, beim Manipulieren von Objekten in der Distanz? Also Objekte, die nicht in deiner Reichweite waren.

T3: Ja, also das hängt zusammen mit dem anderen. Also ich fand das sehr gut, obwohl die weit weg sind, dass ich die eigentlich recht präzise dann noch platzieren konnte, obwohl sie eigentlich weit weg sind. Also gerade zum Beispiel diese Nase auf den Kopf setzen und sowas, das ist ja eigentlich gar nicht so easy auf die Distanz und das fand ich eigentlich echt schön. Ich habe mir nur da nur manchmal gewünscht, aber das hat weniger mit der Manipulation an sich zu tun, dass man irgendwie noch eine Seitenansicht hätte. Also man kann von den Schatten nicht darauf schließen, wie nah man jetzt an dem Objekt ist. Da wäre dann auch noch so eine Sache, es könnte interessant sein, dass man - nur so eine Idee - wenn man jetzt das eine Objekt schneidet, das andere, dass man dann auch noch mal taktiles Feedback irgendwie bekommt. Das man jetzt sagt irgendwie jetzt überschneiden sich die Objekte. Weil oft, zumindest bei den Aufgaben die wir jetzt hatten, ging es ja darum Objekte aneinander auszurichten und nicht ineinander zu schachteln. Und wenn man da jetzt noch mal so ein bisschen Feedback geben würde, im Sinne von, okay du bist jetzt genau an der Kante und jetzt bist du im Objekt, jetzt brummt es da die ganze Zeit - das wäre vielleicht eine Idee, um das zu erleichtern.

I: Auf jeden Fall keine schlechte Idee. Wie würdest du die Komplexität von dieser Applikation bewerten? Würdest du sagen, es war eher komplex zu verwenden? T3: Also ich finde es, dafür was man damit machen kann, also sagen wir es mal so: 3D-Modellierung ist ein komplexes Problem, das ist einfach keine einfache Geschichte. Und dafür fand ich es eigentlich ein sehr simples Interface, dafür dass man doch eigentlich diese ganzen Grundthemen der 3D-Modellierung abhakt. Klar, es hat eine Lernkurve, man muss es lernen, aber ich fand es jetzt insgesamt durchaus durchdacht und durchaus angenehm, also schon eine angenehme Variante. Also insgesamt ein positiver Eindruck. Wie du merkst ich sage die ganze Zeit, ich habe mich trottelig angestellt [lacht] ich habe das Gefühl, ich hätte es noch besser machen können.

I: Naja, aber wenn man das erste Mal irgendwas verwendet, gerade wenn du weniger Erfahrung mit VR hast.

T3: VR habe ich super wenig, ja das fängt jetzt erst an als Thema interessant zu werden und gerade mit dem Christoph Anthes.

I: Gut, letzte Frage: Wie würdest du beurteilen, wie präzise hast du mit der Applikation arbeiten können?

T3: Ja, deswegen habe ich das vorher erwähnt, so Feedback wäre ganz

schön gewesen. Ich hatte schon oft das Gefühl, also ich richte auch in der realen Welt Sachen oft ziemlich genau aus, weil ich einfach so einen Tick habe, dass Sachen in einem rechten Winkel zueinander sind, also nicht immer, aber manche Sachen halt. Und da hatte ich jetzt schon das Gefühl, dass mir das noch nicht genug Kontrolle bietet. Also theoretisch schon, dadurch dass ich halt den Rotationswinkel einstellen kann. Aber so ein bisschen mehr snap to grid, also ein bisschen mehr Magnetismus in dieser Welt, sodass ich wüsste, so jetzt sind sie wirklich Steg an Steg, sauber, genau ausgerichtet, das hätte ich mir vielleicht schon auch noch mehr gewünscht.

I: Okay, gut, das war es mit den Fragen eigentlich. Wenn du noch irgendetwas anmerken möchtest, kannst du das gerne jetzt machen.

T3: Ich glaube, da ist echt einiges an Gehirnschmalz da rein geflossen, also sich da mal zu überlegen, wie man die Technik macht, das finde ich schon cool, coole Sache, gutes Projekt auf jeden Fall.

I: Danke! Dann wäre es das gewesen, danke für deine Zeit!

T3: Gerne!

Appendix B

Contents of the CD-ROM

Format: CD-ROM, Single Layer, ISO9660-Format

B.1 PDF files

Pfad: /

Krautgartner_Lorenz_2017.pdf Master Thesis

Pfad: /Online_Sources

Nielsen.pdf [24]

B.2 Project Executables

Pfad: /Project

promt.exe Executable for use case application
containing PROMT.

Prompt Source folder containing *Unity* project files.

B.3 Other

Pfad: /Images

*.jpg, *.png Original Rasterbilder

References

Literature

- [1] Eric Allen Bier and Maureen C. Stone. “Snap-dragging”. *ACM SIGGRAPH Computer Graphics* 20 (1986), pp. 233–240 (cit. on pp. 27, 28, 34).
- [2] Doug A. Bowman and Larry F. Hodges. “An Evaluation of Techniques for Grabbing and Manipulating Remote Objects in Immersive Virtual Environments”. In: *Proceedings of the 1997 Symposium on Interactive 3D Graphics*. I3D '97. Providence, Rhode Island, USA: ACM, 1997, 35–ff. (Cit. on pp. 15, 16).
- [3] Doug A. Bowman et al. *3D User Interfaces: Theory and Practice*. Second. Redwood City, CA, USA: Addison Wesley Longman Publishing Co., Inc., 2017 (cit. on pp. 4, 12, 27–29, 36).
- [4] Brookshire D. Conner et al. “Three-dimensional Widgets”. In: *Proceedings of the 1992 Symposium on Interactive 3D Graphics*. I3D '92. Cambridge, Massachusetts, USA: ACM, 1992, pp. 183–188 (cit. on pp. 9, 10, 27).
- [5] T. Duval, A. Lecuyer, and S. Thomas. “SkeweR: a 3D Interaction Technique for 2-User Collaborative Manipulation of Objects in Virtual Environments”. In: *Proceedings of the 2006 IEEE Symposium on 3D User Interfaces*. Mar. 2006, pp. 69–72 (cit. on p. 19).
- [6] S. Frees and G.D. Kessler. “Precise and rapid interaction through scaled manipulation in immersive virtual environments”. *IEEE Proceedings. VR 2005. Virtual Reality, 2005*. c (2005), pp. 99–106 (cit. on pp. 19, 33).
- [7] Scott Frees, G. Drew Kessler, and Edwin Kay. “PRISM Interaction for Enhancing Control in Immersive Virtual Environments”. *ACM Transactions on Computer-Human Interaction Article 2 (May 2007)*, 31 pages. 14.1 (May 2007), p. 31 (cit. on pp. 20, 38).

- [8] Bernd Fröhlich and John Plate. “The Cubic Mouse: A New Device for Three-dimensional Input”. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '00. The Hague, The Netherlands: ACM, 2000, pp. 526–531 (cit. on p. 7).
- [9] Stephanie Houde. “Iterative design of an interface for easy 3-D direct manipulation”. *Proceedings of the SIGCHI conference on Human Factors in Computing Systems* (1992), pp. 135–142 (cit. on pp. 1, 2, 25).
- [10] Ryan P. McMahan et al. “Separating the Effects of Level of Immersion and 3D Interaction Techniques”. In: *Proceedings of the ACM Symposium on Virtual Reality Software and Technology*. VRST '06. Limassol, Cyprus: ACM, 2006, pp. 108–111 (cit. on p. 23).
- [11] Daniel Mendes et al. “The Benefits of DOF Separation in Mid-air 3D Object Manipulation”. In: *Proceedings of the 22Nd ACM Conference on Virtual Reality Software and Technology*. VRST '16. Munich, Germany: ACM, 2016, pp. 261–268 (cit. on pp. 9, 13, 25–27).
- [12] Mark R. Mine. *ISAAC: A Virtual Environment Tool for the Interactive Construction of Virtual Worlds*. Tech. rep. Department of Computer Science, University of North Carolina, 1995 (cit. on p. 16).
- [13] Mark R. Mine, Frederick P. Brooks Jr., and Carlo H. Sequin. “Moving Objects in Space: Exploiting Proprioception in Virtual Environment Interaction”. In: *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques*. SIGGRAPH '97. New York, NY, USA: ACM Press/Addison-Wesley Publishing Co., 1997, pp. 19–26 (cit. on pp. 17, 19).
- [14] Thi Thuong Huyen Nguyen and T. Duval. “Poster: 3-Point++: A new technique for 3D manipulation of virtual objects”. In: *2013 IEEE Symposium on 3D User Interfaces (3DUI)*. Mar. 2013, pp. 165–166 (cit. on p. 21).
- [15] Jakob Nielsen. *Usability Engineering*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1993 (cit. on p. 46).
- [16] Jeffrey S. Pierce, Brian C. Stearns, and Randy Pausch. “Voodoo Dolls: Seamless Interaction at Multiple Scales in Virtual Environments”. In: *Proceedings of the 1999 Symposium on Interactive 3D Graphics*. I3D '99. Atlanta, Georgia, USA: ACM, 1999, pp. 141–145 (cit. on pp. 17, 18).
- [17] Ivan Poupyrev et al. “The Go-go Interaction Technique: Non-linear Mapping for Direct Manipulation in VR”. In: *Proceedings of the 9th Annual ACM Symposium on User Interface Software and Technology*. UIST '96. Seattle, Washington, USA: ACM, 1996, pp. 79–80 (cit. on pp. 14, 15).

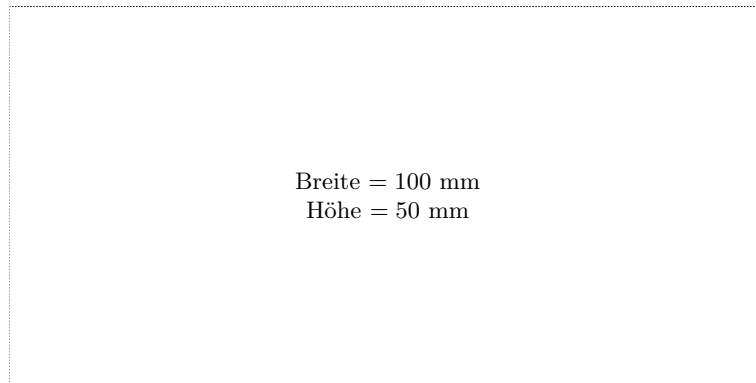
- [18] Peng Song et al. “A Handle Bar Metaphor for Virtual Object Manipulation with Mid-air Interaction”. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '12. Austin, Texas, USA: ACM, 2012, pp. 1297–1306 (cit. on pp. 21, 27).
- [19] Richard Stoakley, Matthew J. Conway, and Randy Pausch. “Virtual Reality on a WIM: Interactive Worlds in Miniature”. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '95. Denver, Colorado, USA: ACM Press/Addison-Wesley Publishing Co., 1995, pp. 265–272 (cit. on pp. 13, 14).
- [20] Ivan E. Sutherland. “A Head-mounted Three Dimensional Display”. In: *Proceedings of the December 9-11, 1968, Fall Joint Computer Conference, Part I*. AFIPS '68 (Fall, part I). San Francisco, California: ACM, 1968, pp. 757–764 (cit. on p. 6).
- [21] Robert Wang, Sylvain Paris, and Jovan Popović. “6D Hands: Markerless Hand-tracking for Computer Aided Design”. In: *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*. UIST '11. Santa Barbara, California, USA: ACM, 2011, pp. 549–558 (cit. on p. 27).
- [22] Matthias M. Wloka and Eliot Greenfield. “The Virtual Tricorder: A Uniform Interface for Virtual Reality”. In: *Proceedings of the 8th Annual ACM Symposium on User Interface and Software Technology*. UIST '95. Pittsburgh, Pennsylvania, USA: ACM, 1995, pp. 39–40 (cit. on p. 16).
- [23] Shumin Zhai and John W. Senders. “Investigating Coordination in Multidegree of Freedom Control II: Correlation Analysis in 6 DOF Tracking”. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* 41.2 (1997), pp. 1254–1258 (cit. on p. 6).

Online sources

- [24] *Thinking Aloud: The 1 Usability Tool*. Jan. 2012. URL: <https://www.nngroup.com/articles/thinking-aloud-the-1-usability-tool/> (cit. on pp. 46, 80).

Messbox zur Druckkontrolle

— Druckgröße kontrollieren! —



— Diese Seite nach dem Druck entfernen! —