Saarland University

Faculty of Mathematics and Computer Science
Department of Computer Science

Bachelor Thesis

Exploring Mid-air Interaction for Menu Control in Virtual Reality with regard to Task Performance and User's Preference

Jan Ehrlich

s9jaehrl@stud.uni-saarland.de

October 30, 2017

Advisor
Marco Speicher

Supervisor
Prof. Dr. Antonio Krüger

Reviewers
Prof. Dr. Antonio Krüger
Dr. Anusha Withana

Erklärung

Ich erkläre hiermit, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.

Statement

I hereby confirm that I have written this thesis on my own and that I have not used any other media or materials than the ones referred to in this thesis

Einverständniserklärung

Ich bin damit einverstanden, dass meine (bestandene) Arbeit in beiden Versionen in die Bibliothek der Informatik aufgenommen und damit veröffentlicht wird.

Declaration of Consent

I agree to make both versions of my thesis (with a passing grade) accessible to the public by having them added to the library of the Computer Science Department.

Saarbrücken,________________________ ________________________________
(Datum/Date) (Unterschrift/Signature)

Abstract

Nowadays, VR is more accessible than ever and the software competition is huge. While topics like motion sickness or virtual presence are well investigated, basic topics as VR UIs are lagging far behind. The greatest problem here is the absence of haptic feedback. Users can not feel with which element they interact and developers cannot prevent users from doing specific actions (e.g. grabbing through walls).
In this thesis we present and compare three different VR interface approaches to mid-air menu-control through Hand Interaction: A physical 3D Interface combined with non-isomorphic techniques to create pseudo feedback, a stiff 2D Interface similar to a typical desktop interface and a Finger Count Interface which enables the interaction with the menu by extending a certain number of fingers.
Through user studies, we found out that the 3D Interface has the best usability and the 2D Interface the best performance. We observed that the Finger Count Interface is unsuitable for VR Menu Control. During development, we faced many unobvious problems and findings, which we summed up in some guidelines.

Contents

1 Introduction

1.1 Motivation

Nowadays, the majority of the population is familiar with PCs and their WIMP (Windows, Icons, Menus, Pointer) system or with touchscreens like smartphones and their Graphical User Interface (GUI) [40]. Menu metaphors are widely established and understood by the users. For example, we have menu buttons representing real buttons, toggle sliders and checkboxes that are just a metaphor for a simple on/off switch or sliders which are the equivalent of a real slider similar to the ones in DJ mixers. In Virtual Reality (VR), we can observe a similar trend. Many planar User Interfaces (UIs) are used for menu control. Often it gets combined with a raycasting technique for selecting interface elements. But „[we] cannot assume that simply transferring conventional interaction styles will lead to usability“ [1] and „[simply] adapting 2D desktop-based widgets is not the ultimate solution“ [2]. The huge benefit of VR is having stereoscopic vision. So it is well suited for displaying three-dimensional objects. So why are planar menus used so often in VR? Instead of interacting with metaphors of objects, one could also just interact with the object itself. For this reason, we developed and compared the two extrema: A planar 2D interface which does not use stereoscopic vision at all and a 3D interface which utilizes the source of many menu metaphors. To dive into the virtual world, we used an HTC Vive[2] and the Leap Motion[4] as the hand tracking input device. Users can interact with both interface concepts by direct manipulation. The main goal of this thesis is to analyze both UI concepts and an additional Finger Count interface which emerged during the study of related works. We want to explore their strengths and weaknesses and see which UI is the most suitable one for menu control.

[1] Bowman et al. [14]
[2] Bowman et al. [14]

1.2 Research Questions

From our motivation, we can derive following research questions:

1. *What is the difference in usability and performance regarding the three UI approaches: Planar 2D UI, 3D UI and Finger Count UI?*
2. *Which options do we have regarding interaction with those menus?*

Since the planar 2D UI is widely applicated in VR applications for menu control, it is essential to find out whether its frequent occurrence is justified or whether there are far better approaches for this task. Menu control is part of the VR user experience just as any other part of a VR application. So it should not only be a means to an end but it should also contribute to the overall user experience while still being effective regarding performance. To find out which UI approach is best suited to fulfill this criterion, a within-subject study was conducted. This study consisted out of two experiments which were used to measure usability and performance, respectively. In the first experiment, the participants should use all three UIs to perform menu selection tasks. For the second experiment, a realistic system menu setup and environment were used. Participants should fulfill realistic menu tasks, like altering the volume or graphics quality, with each of the three UI approaches. During the study, we measured dependent variables like completion time, accuracy and error rate to estimate the performance of the individual UIs. To measure usability, standardized tests were used. The User Experience Questionnaire (UEQ), the NASA Task Load Index (NASA-TLX) and the Motion Sickness Assessment Questionnaire (MSAQ) were some of them.

1.3 Significance of the Studies

Research cannot keep up with the rapid growth of the VR consumer market. Companies create VR applications and make design decisions, without validated scientific foundation. As we already stated, one can not assume that laws of 2D interfaces will also apply for VR interfaces. Still, many essentials are not investigated in the context of VR. Therefore, we wanted to do research one of those topics. Is it ok to use planar 2D UIs? Are there better alternatives? What is more important: Affordance or Familiarity? Additionally, the conducted study should serve as a baseline for performance of UIs for menu control in VR. An analysis of the collected data showed that both, the 2D and the 3D interface, achieved pretty good usability ratings through the UEQ, NASA-TLX and MSAQ. A tendency in favor of the 3D Interface is noticeable here. On the other hand, the 2D UI was slightly better than the 3D UI regarding performance, with both of them providing acceptable results. Unfortunately, the promising Finger Count UI could not meet the requirements for a viable menu control interface, neither regarding usability nor performance.

1.4 Outline

Throughout the Related Work chapter [see chapter 2], many basic concepts of VR interactions will be presented and clarified. It also shows the effect of a rudimental haptic feedback device on user performance in VR. With this knowledge, it is possible to estimate how much of a difference the lack of haptic feedback makes. Other VR User Interfaces will be discussed and how they helped in setting many parameters of our UI before the actual development. The chapter includes a related work, which showed how to support the user in estimating distance during the lack of haptic feedback or in other words while touching the void. And finally, the Finger Count interaction will be presented and discussed in this chapter. The following Concept chapter [see chapter 3] documents the available UI elements and the widget which were created with them. Also, it concerns menu structure and the used virtual environment. The Implementation chapter [see chapter 4] shows how and with the help of which tools, the individual UI elements were created. Afterwards, the User Evaluation [see chapter 5] chapter provides all important information about the conducted study, e.g. apparatus, participants, study design, procedure but also the results followed by a discussion of the results. During the whole work, many aspects were documented and the most important ones were documented in the form of guidelines, which can be found the eponymous chapter Design Guidelines for Mid-air Menu Control [see chapter 6]. The Conclusion chapter [see chapter 7] will sum up the most important aspects of this thesis and answer the initially asked research questions. Finally, the Future Work chapter [see chapter 8] gives an overview of what is possible and interesting to research in the future. It also contains a huge collection of UI ideas which did not make it into our prototype.

2 Related Work

2.1 Haptic in Virtual Reality

The lack of haptic feedback in Virtual Reality is a major issue and many researchers are working on ways to fill this gap [15]. One solution approach could be the integration of devices that simulate haptic feedback. For example, we have Dexmo: An Inexpensive and Lightweight Mechanical Exoskeleton for Motion Capture and Force Feedback [27]. It is a hand-worn exoskeleton device that is capable of measuring finger positions and stopping them from moving. Dexmo's inexpensiveness and lightness are solved by dropping expensive motors, complex transmissions or sensor modules. It only needs a small shifting servo to lock a joint resulting in a freeze of the finger's movement. The lock system consumes very little power for the small and efficient sensors to track the finger positions and for switching the joints from the locked to the unlocked state, similar to the E-Ink system in the Kindle Paperwhite and its half white-and-black dots which just turn around. But this design choice ends up being a tradeoff for the user experience. When touching a tough object in the virtual world, for example a stone, you will get the most rigid feeling possible. On the other hand, you cannot simulate soft objects like a rubber duck. Combining the lock system with a motorized variable force feedback system could get rid of the trade-off. Nonetheless in an evaluation of the performance of the system, while only observing the error rate of 20 participants, they got following results: The average error rate without force feedback was 61%. With the use of force feedback, they could lower the error rate to 44%, which shows a significant improvement. Also, the participants' informal responses were mainly positive.

A problem of the exoskeleton based system is the need of a fixation point. In the Dexmo system, you had the hand as a fixation point to restrict the fingers but you cannot restrain the hand movement. You would need a bigger and more powerful system, for example on ones back, to do so. But then you cannot restrict the whole body movement without a huge, bulky fixation point mounted in the real world. Such a system would be expensive, heavy, space consuming and therefore unfriendly

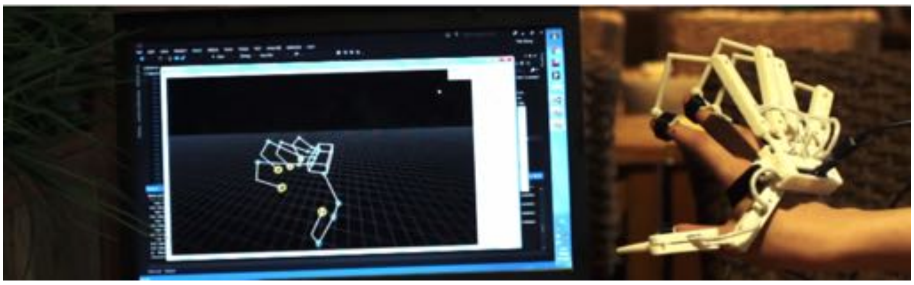


Figure 2.1: An early and basic Dexmo Prototype [27]

towards average consumers like AxonVR [1] is showing us.

Figure 2.2: An early concept render of AxonVR [1]

This restriction is critical because as Bowman et al. [16] say "3D UI should maintain spatial and temporal correspondence between multiple feedback dimensions that the user receives. For the sensory dimension, for example, if the visual feedback conflicts with the kinesthetic or proprioceptive feedback generated by the body, then user performance rapidly degrades". Such a conflict can occur if a virtual hand is not allowed to pass through objects, but the real hand keeps on moving through it. This behavior is often called feedback displacement.
But maybe we do not need restriction or feedback to such an extent. Maybe it is possible to get similar performance without full-body haptic feedback with the help

of nonisomorphic approaches or like Bowman et al. are also calling it: *magic virtual tools*.

2.2 Isomorphic vs. Nonisomorphic approach

„The design of 3D manipulation interfaces has been strongly influenced by two opposing views“ Bowman et al. [17]. On the one hand, we have the isomorphic view, which suggests a one-to-one mapping between hand motion in the physical and virtual worlds, just on the grounds that it is the most natural way of interaction and therefore better for users. Through studies, Knight [33] indicates that isomorphism is indeed more natural, but it also buries some problems: An impractical mapping because of constraints in the input device like tracking range and the limitations through the human body like the arm length that naturally limits the reaching distance. On the other hand, we have the nonisomorphic view, that recommends trying not to imitate the physical reality and instead, create mappings and interaction techniques that are specifically tailored to 3D environments to achieve a better version of reality (e.g., Stoakley et al. [46]). So it is comparable to our world which is limited and ruled by the laws of physics and a new world where you can define the rules just as you wish. But this also comes with the risk of creating unfamiliarity, confusion, and misunderstandings for the user.
Therefore we try to find a suitable balance between the isomorphic and nonisomorphic approach to gain the most out of both approaches for our work. We decided to place our project's user interface in arm's reach and use a reliable hand-tracking device to get the most natural way of interaction. The Leap Motion [4] should accomplish this for us. To address the issue of lacking feedback we try to use nonisomorphic techniques to create an illusion of feedback and thereby enhance the overall user experience.
For example, in our implemented prototype we created 3D Buttons [see chapter 3.4]. Those buttons are solid. So they can be touched. But instead of letting the hand pass through the button, the button gets pushed by the hand along a defined axis. The button can be moved as far away from its original position as the user wants. But the button itself attempts to get back to this position, similar to an elastic spring that gets expanded and desires to recover its original state or a balloon that gets pushed under water and rises above the water again. So we combined an isomorphic approach, that one's hand does not slip through a UI element and a nonisomorphic approach, that the UI element is floating in space and can penetrate

a trigger element.

2.3 3D Manipulation Tasks

To analyze and compare as many interactions as possible, we need to break them down into the most basic tasks that they have in common. For 3D manipulation Bowman et al. [18] proposed following tasks:

- *Selection*: To pick or identify a particular virtual object from a set of all available objects. The real-world equivalent would be choosing an object by hand.
- *Positioning*: Altering the location of a virtual object. The real-world equivalent would be moving an object from one position to another.
- *Rotation*: Altering the orientation of a virtual object. The real-world equivalent would be rotating an object in space.

Those tasks are also part of User Interfaces and therefore also essential for our work. All Interface elements need to be at least selectable. Some of them are also movable or rotatable, for example slider or our 3D wheels.
Each of those most basic tasks comes with a huge variety of variables which influence user performance and usability significantly. Poupyrev et al. [42] investigated a lot in the topic of those *task parameters*. They did not enumerate every possible and existing parameter there is, but they illustrated the most prominent ones according to their experience and related work.
Selection Task Parameters:

- Main parameters:
  - Amount of selectable objects
  - Distance to the target
  - Target size
  - Direction to the target relative to the user
  - Amount of target occlusion
- Minor parameters:
  - Target dynamic

  - Density of other selectable objects near the target

Position Task Parameters:

- Main parameters:
  - Distance to the object that should be moved
  - Direction to the object that should be moved
  - Distance to the desired target location
  - Direction to the desired target location
  - Required precision for the positioning task
- Minor parameters:
  - Visibility of the target location
  - Dynamics of the target location
  - Occlusion of the target location
  - Size of the object that should be moved

Rotation Task Parameters:

- Main parameters:
  - Distance to the object that should be rotated
  - Direction to the object that should be rotated
  - Original orientation
  - Target orientation
  - Required precision for the rotation task

Poupyrev et al. [42] determined user performance under the aspects of task completion time, accuracy, error rate, memorability, learnability and immersion.
For every design decision, we need to consider each of those parameters precisely and how altering them will affect the user performance. Also, those parameters correlate to each other. E.g., if we increase the target size of our interface elements, we will either increase the distance to the target or the density of other selectable objects around the UI element. To get more information on already examined parameters and their optimal values in UIs, we looked at other VR User Interfaces and gathered insights which we can use in our later work.

2.4 VR User Interfaces

2.4.1 An Intuitive VR User Interface for Design Review

Knöpfle & Vöß [34] faced the problem of improving the development cycle of the automotive industry. Physical prototypes are essential there, but also very cost and time intensive. So the overall goal for them was to reduce the needed amount of those prototypes until the finished product. Traditionally drawing or CAD models helped regarding this issue, but they have some huge problems, which forced the manufacturers to keep on using Physical Prototypes. Drawings are still time-consuming to produce, hard to create changes and they only provided a particular view of the product. CAD Models were faster to create and modifications are easier, but also suffered from the problem of the limited viewing angles. So Knöpfle et al. wanted to create a VR User Interface which helps experts to review, discuss and judge a design.
Knöpfle et al. identified the same atomic tasks we mentioned in chapter 2.3 3D Manipulation Tasks with the addition of text input and quantification. As an input device, they used a self-developed remote called the *Flystick*. This decision was based on the lack of viable alternatives.

Figure 2.3: The used Flystick [34]

They solved the Position [see 2.3] and Rotation task [see 2.3] by basically grabbing the virtual world and move/rotate it around the grabbing point. The selection

task was split into two different selection types. They differentiated between object selection and menu selection. For object selection, they used the *spotlight technique* firstly introduced by Liang & Green [38].
To interact with the selectable objects, a menu was essential. It is part of any system which provides a huge variety of functionality, in order of being able to switch between the functions. In their first approach, they tried to transfer a layout similar to a desktop user interface into the virtual world. It strongly resembled the Windows(tm) menu system. This design choice was purely made out of their hypothesis, that an already well-known system will achieve better learnability. But they were appalled to discover, that the aspect of familiarity was by far not enough, to make up for the lack of usability. The users had an immense mental demand and needed to aim far too precisely, to just select a menu element. So Knöpfle et al. realized very early, that their menu system should forgo high accuracy input provided by hand translation.
In their second iteration, they used a pie menu, firstly introduced by Callahan et al. [22]. With this menu, they solved their accuracy problem and provided equal selection time for all menu elements.

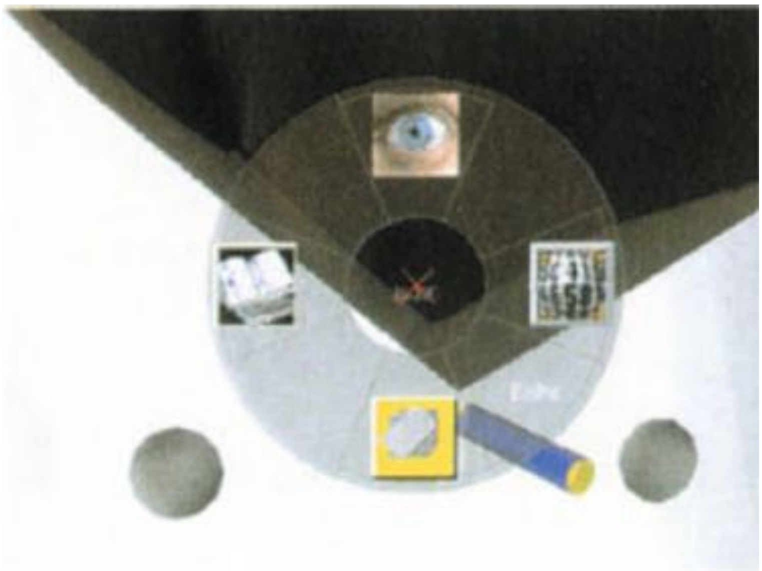


Figure 2.4: The used pie menu [34]

For the additional quantification task, they used a VR jog dial. The real-world counterpart is a relict from the 1970s until the late 1990s, where it was often used in video recorders to fast-forward or fast-backward. The jog dial, both in the real world and in Knopfle et al.'s implementation, are rotatable from their original position to

+/- 90 degrees. On rotating the dial into a positive position, the viewed material gets fast-forwarded and in a negative position, the material gets fast-backwarded. To not face the initial accuracy problem, they partitioned the virtual jog dial into five different zones. A neutral zone where nothing happens, a fast-forward and fast-backward zone and a very fast-forward and very fast-backward zone.

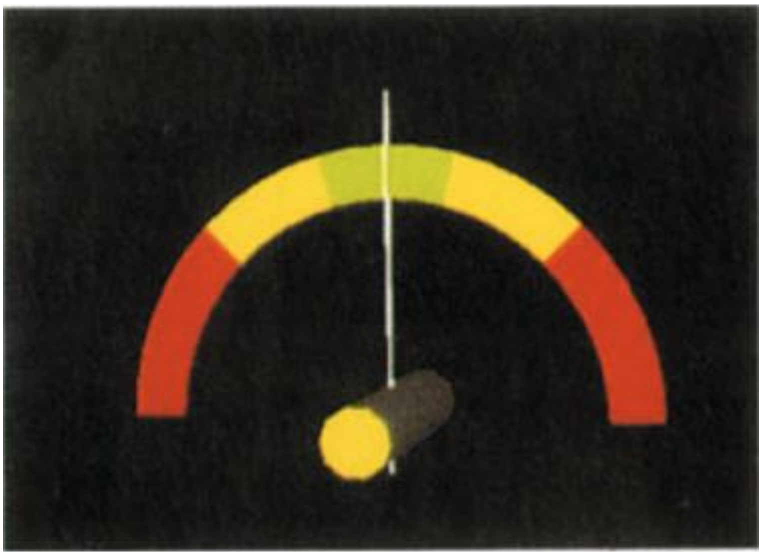


Figure 2.5: The used jog dial [34]

The text input task could not be solved in *Virtual Reality*, so they shifted the task onto a separate PC near the VR system.
Again, we can see that just transferring a given system from one medium to another, is not the optimal solution regarding usability. Familiarity is a factor that needs to be concerned, but familiarity does not justify bad usability at all. Until it is possible to work in VR as precisely as in the real world, it is essential to come up with new ways of interaction, which support the users in achieving their goals. In our prototype, we decided to make all elements big enough in order of being able to interact with them with the whole hand and not just with a single finger.

2.4.2 The Personal Cockpit: A Spatial Interface for Effective Task Switching on Head-Worn Displays

Ens et al. [24] invented a system which creates virtual 2D windows around the user with which he can interact through direct manipulation [see Figure 2.6]. Their primary focus was on enhancing the switching time of mobile application. They conducted a design-space exploration with many studies, which helped them to find the parameters for optimal performance, e.g. distance and angle from user to windows. But even before executing them it was already possible to narrow down some variables. Hezel & Veron [31] stated that the human eyes' accommodation and convergence allow to comfortably view objects starting at a distance of about 0.25 meters. All windows should have the same distance to the user, because of an already investigated decrease in performance when separating information in the visual field by depth. Although it is worth mentioning, that the performance drop was only about 10% in the conducted studies [47].

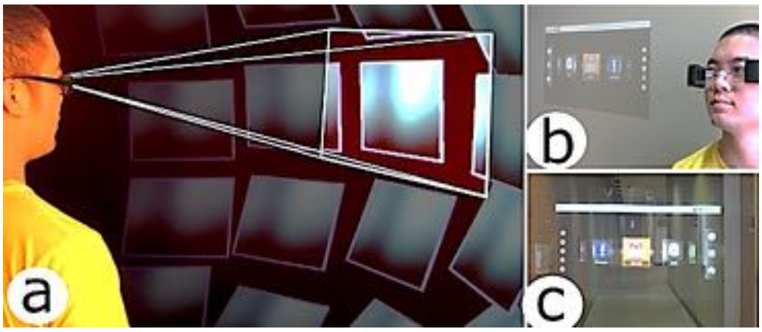


Figure 2.6: The concept of the Personal Cockpit [24]

Ens et al. estimated, that interactive objects should be between 50 to 60cm in front of the user and 70 to 80cm away from his dominant side. Those estimations are based on NASA's *Man-System Integration Standards* [39] and their documentation regarding average maximum arm reach. At the given time of their work, they faced technical problems regarding reliable tracking and motion stabilization, which is described as essential, if you want to achieve the most natural feeling during direct manipulation. Only like this, it is possible to occlude the virtual objects with one's hand. So they used a *CAVE setting* to emulate a Head Worn Display.
For our work, we are in the favorable position to solve the problem as intended, by using the Leap Motion [4].
With the help of Ha et al. [28] work regarding workspaces in multi-display environments, Ens at al. were able to narrow down possible multi-display arrangements to 4 different options:

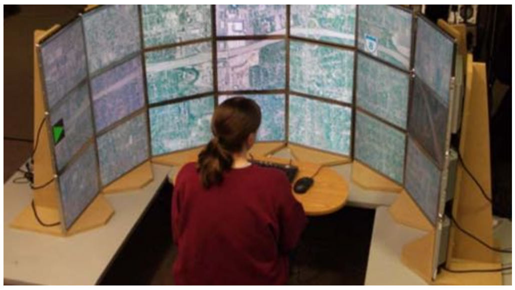
- *World-fixed*: The virtual displays are placed at a distinct location in the virtual world or are attached to an object which is part of the virtual world.
- *View-fixed*: The Displays are attached to the user's head and follow its move-

ment.

- *Body-fixed*: The Displays are attached to the user's torso and follow its movement.
- *Hand-fixed*: The Displays are attached to the user's hand/arm and follow its movement.

Also, the assumption was made, that for their system, a curved layout of displays would be better than a flat layout, based on the work of Shupp et al. [44]. It is just an assumption because Shupp et al. examined the effects of viewport size and curvature in the context of geospatial tasks, like searching, route tracing and comparing data of different maps. And even in their experiment [see Figure 2.7], they observed that effects were more or less present depending on the task. But for all easy tasks of their experiment, they could figure out, that there is a performance increase regarding speed when using a curved display instead of a flat one, no matter the screen size.

(a) Flat Monitor Layout



(b) Curved Monitor Layout

Figure 2.7: Shupp et al. [44]'s Experimental Setup

So we also use a curved layout in our work. Compared to the tasks given in Shupp et al.'s experiments, our tasks are rather easy regarding searching and analyzing the given material. In our case there is also the huge advantage of the curved layout: It is reachable just by arm movement and rotating the body. If the flat layout expands too much, users would need to reposition their whole body to interact with elements which are far away. Shupp et al. also observed this behavior and they conclude that this type of physical navigation is superior and that it enhances the visual access and processing of imagery.
In early stages of our project, we had to decide whether we should use "multi-display menus". We could either display one level of menu depth at a time by having one

imaginary display, where we just replace the content or we could display all levels of menu depth by creating multiple imaginary displays. Shupp et al. helped us deciding here. They determined that frustration was significantly decreased by using larger screens than just their single monitor setup which they linked to the better use of the human visual capacity and the superiority of physical navigation over virtual navigation (on monitors). By using the whole body to navigate it is enormously less probable to make errors and that also corresponds with the observed reduction of frustration in the system which affords physical navigation.
As stated at the beginning Ens et al. conducted a design-space exploration to verify and find the right values for the parameters. In the following, I will summarize their results and all important aspects regarding each parameter.

Single Display Viewing Size and Distance

The optimal display size for their Personal Cockpit depends on the display width to Field of View ratio. The optimal task time was achieved when the display was around $\frac{3}{4}$ the size of the Field of View, probably because of the reduced head motion. Task performance was not affected by the display distance, but the nearest display distance (40cm) was responsible for an increase of discomfort among the participants. So the display distance is still just limited by the users average reach.
For our work, those outcomes are not as usable as for Ens et al. Their primary task is application switching and identifying single applications as fast as possible increases their usability a lot. In our work users need to interact with elements which are grouped by context. And as stated in the first UI of 2.4.1, we need to make these elements big enough to interact with them with the whole hand which will require a certain size. If it is possible to narrow down a level of menu depth to around $\frac{3}{4}$ the size of the Field of View without sacrificing the usability of individual elements, we will do so. Otherwise, we will not.

Single Display Interaction Position and Distance

Where the first parameter focused on visual output, the second one concentrates on direct input. Therefore they conducted a study regarding the effects of the four previously introduced multi-display arrangements and display distance on task completion time, pointing errors and fatigue ratings. While there was no difference in task completion time neither for different arrangements nor for different distances, the world-fixed arrangement was the clear winner regarding precision. The other had

a fundamental flaw: The users moved the displays unintentionally while trying to interact with them. Surprisingly also the distance of the display affected pointing errors. The error rate was greatest at 60 cm and lowest at 50 cm.
By this, we can follow, that the optimal distance to an interactable element for both, input and output, should be around 50cm. For our work, we also decided to use the world-fixed arrangement. It would be possible to solve the large drawbacks of the other arrangements by defining a threshold, so the interface does not move unintentionally that quickly. But investigating the exact parameters of such a system would be a possible topic for future work and goes beyond the scope of this bachelor thesis.

Multi Display Layout: Focal point and angles

The curved layout of the displays needs a focal point. For Ens et al. [24] the two possible focal points were either around the center of the user's body or around the shoulder of the user's dominant hand. These different focal points did not affect completion time, but the error rate was reduced significantly when the focal point was around the shoulder of the user's dominant hand. As an optimal layout Ens et al. proposed following Figure [see Figure 2.8].

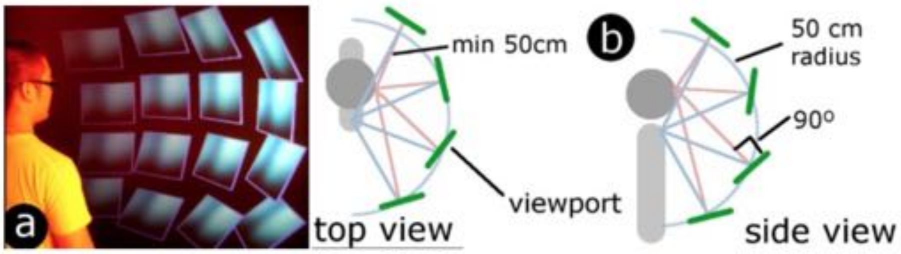


Figure 2.8: The final design of Ens et al.'s Personal Cockpit [24]

In our Prototype, we will also use a focal point around the shoulder, but we will not arrange the menu in the form of a sphere as in Figure 2.8b. Partitioning a menu into very small sections is hard. It would weaken the grouping and therefore it would be harder to understand which elements belong to the same menu level. Also, the size of components varies too much, so it is not possible to fit every element in an imaginary window of the same size, without sacrificing usability. For our work, it is also important to mention, that the height in which the system will be located, is depending on the user's height.

With all the collected data Ens et al. were able to create a system which is 60 % faster in task switching than the other two used baseline interfaces.

2.5 Feedback in 3D User Interfaces

Bowman et al. [16] state that feedback is essential. Not only in VR, but also in 2D GUIs or even for trivial things, like doorknobs or physical buttons on keyboards (mechanical-switch keyboards vs. dome-switch keyboard). Feedback refers to any information that helps the user understand the current state of the system, the results and effects of his performed operations and the state of ongoing tasks. This information could come from the system itself, the environment or even the user's body. There are many dimensions of feedback, so it is useful to narrow it down to major classifications. Bowman et al. proposed following two most basic classifications:

- *Sense-based Feedback*: A feedback dimension consisting of visual, auditory, tactile and olfactory feedback which is created outside the human body. But it also consists of proprioceptive and kinesthetic feedback produced inside the human body, which is responsible for the feel of the position and movement of the body and its limbs. This feedback can be consciously prepared and controlled by the UI designer, except for the proprioceptive and kinesthetic feedback. Bowman et al. also state that providing compliant feedback on multiple channels of senses will improve user performance and satisfaction in 3D UIs.

- *System-based Feedback*: Feedback from the systems point of view. Bowman et al. defined this feedback based on a similar definition by Smith & Smith [45] and split it into three different categories:
  - *Reactive Feedback*: All the generated visual, auditory, haptic and proprioceptive feedback created by interacting with the system.
  - *Instrumental Feedback*: Feedback which is created during the use of the tools and controls of the system. E.g., the vibration when moving a mouse over an uneven surface or the drag when pulling a lever.
  - *Operational Feedback*: Feedback which presents the user the effects of his actions.

So in our project, the reactive feedback would be the color change of a button, when actually pressing it, the sound which is played when pressing a button or moving a slider or even the kinesthetic feedback when moving the arm to interact with an element. Instrumental feedback is completely missing since the user does not need to interact with any physical object besides the HTC Vive. One could argue, that the only available instrumental feedback could be the awareness about the pressure of the HMD when rotating one's head, caused by the centrifugal force [1]. The operational feedback can be perceived when seeing the consequences of triggering a particular element, e.g. turning on a light source or opening a new menu.

Bowman et al. also criticises that the boundaries of those classifications are a bit vague, but still they are a valuable tool to analyze the overall feedback and being able to sync up different feedback in order of being able to enhance the most important aspect of feedback, which would be *compliance*.

As we already stated and discussed in 2.1, sometimes it is not possible to provide all types of feedback. Especially when not being able to provide haptic feedback, a feedback substitution principle has often been used. Bowman et al. stated, that for this absence, additional audio or visual cues can be utilized instead. For example, in a selection task, when touching a virtual object, the object can be highlighted to signalize a successful touch interaction. Butterworth et al. [20] enhanced this selection technique, by predicting the most probable object, which will be touched next. When the user continued his current operation, the object gets selected.

Feedback substitution is a major part of our work. One of our primary goals is to find whether and how much of a difference there is, between a system with the bare minimum of feedback substitutions and a system with additional feedback substitution.

Chan et al. [23] researched direct-touch interaction on intangible displays. They also used different feedback substitution methods to enhance the performance and user experience. Firstly they wanted to know, how big of a problem, the lack of tactile feedback was. Therefore they conducted a study, where participants had to guess the position of a virtual object with their finger. The participants had three seconds, to bring their real finger into virtual contact with a virtual object, which was slightly larger than the width of conventional fingers. After the time ran out, the object disappeared, the current finger position was tracked and the distance from the fingertip to the object was calculated. During the whole task, no feedback was provided at all.

Participants made far fewer errors in the x and y-axis than the z-axis. In fact, thirty percent of the total finger placements were further than 30mm away from the actual

[1] Wikipedia.org: Centrifugal Force

surface along the z-axis.
At this point, it is important to mention that the used device in all of their conducted experiments, was only able to display 2D planar images which did not provide true depth perception like a stereoscopic HMD does. So one cannot be sure whether the observation and results still apply for systems where users are capable of using their actual depth cues.

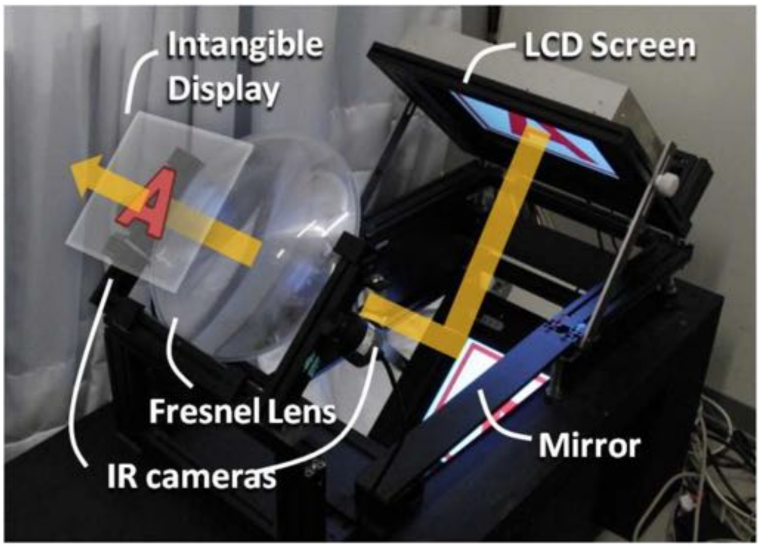


Figure 2.9: The used Device during the Experiment of Chan et al. [23]

After acquiring a baseline, Chan et al. conducted a similar experiment, but this time with visual and audio feedback. The visual feedback was created by using a pseudo-shadow effect which was projected onto the virtual object. The shadow is a very intuitive and self-explanatory tool, an isomorphic approach because it translates a well-known real-world behavior into the virtual world. The closer you get your finger to the surface of an object the closer the finger's shadow gets to the finger [see Figure 2.10].
The position of the shadow could be influencing performance as well. At first, they wanted to investigate three possible positions for the light source: from above-left, from above and from above-right, which would lead to a shadow on the right, left or below the hand. Through a pilot study, they could already exclude the pseudo-shadow placed below the hand, because of the hand occluding the shadow most of the time and thereby render it useless.

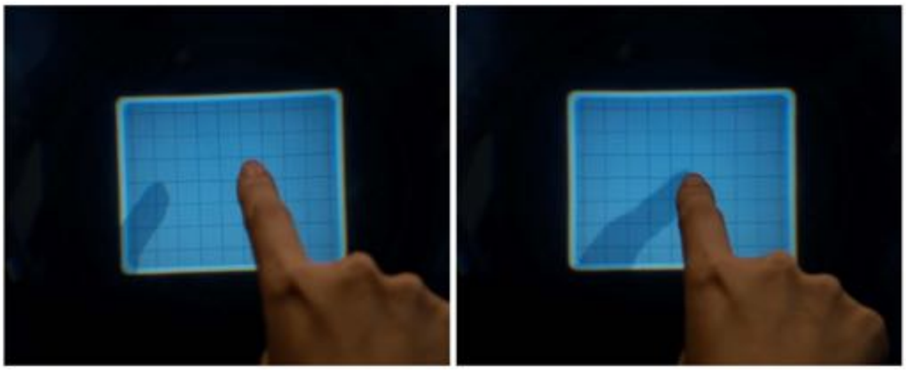


Figure 2.10: The used pseudo-shadow effect from the Non-dominant side [23]

As audio feedback, they used a short non-speech audio sound which was repeatedly played when participants hovered their fingers on the surface.
In the second experiment, participants should place their finger on a marked target and hold it in place for 2 seconds. They used a within-subject design for the experiment, so one participant should do the task with no feedback at all, with audio feedback only, with a pseudo-shadow effect from the non-dominant side and a pseudo-shadow affect from the dominant side. The ordering was counterbalanced and the target positions were randomized.
As a result, the difference of the Non-dominant-side shadow towards the Dominant-side shadow was borderline significance ($p < 0.03$) in favor of the Non-dominant-side shadow. All feedback was useful and improved user performance significantly concerning completion time. Participants should also rank the different feedback approaches after the experiment, but no clear preference was visible here.
By this work, we can see that pseudo-shadows and audio feedback are handy tools to improve user performance while reducing confusion. In our work, we provide audio feedback for all interface elements, which should provide similar performance improvements as pseudo-shadows. We did not use pseudo-shadows in the implemented menu because we wanted to clearly differentiate between the virtual environment and a system menu, which is not part of this environment. Therefore our menu does neither receive nor cast a shadow. But still, it is an interesting question for future work, whether such pseudo-shadows will improve performance during the use of stereoscopic HMDs and which metaphors could be used to integrate those shadows in a menu seamlessly.

2.6 Finger-Based 3D Gesture Menu Selection

Kulshreshth & LaViola [35] investigated the topic of numeric gestures as menu control, which was suggested by Bowman et al. [19] but still unexplored at that time. Finger count gestures require extending a certain amount of fingers in order of being able to interact with an element which is linked to the shown amount. Those gestures contain the tremendous potential to be a natural and intuitive approach for menu selection. A huge benefit of those gestures for menu selection is the fact that they are not affected by Fitts's Law [25].
Kulshreshth and LaViola used a 55" Sony HDTV as an output device and a Creative Interactive Gesture Camera (a depth-sensing camera) as an input device which tracked the hand and finger positions [see Figure 2.11].

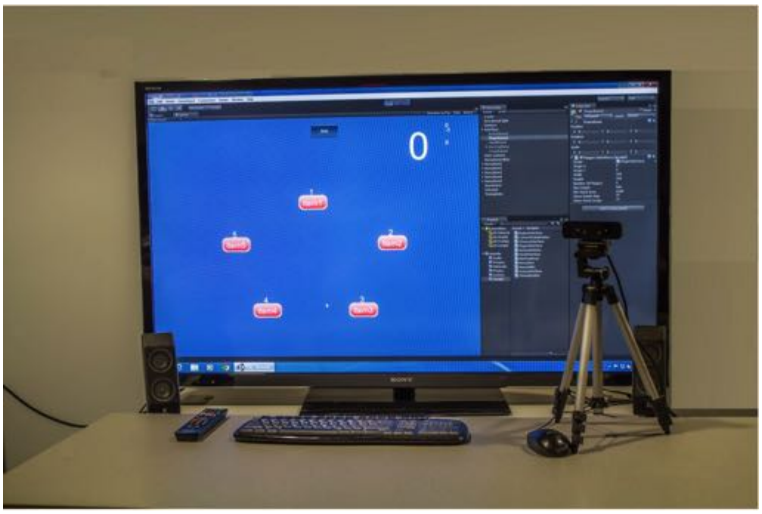


Figure 2.11: The used Experimental Setup of Kulshreshth & LaViola [35]

To compare different menu selection techniques against each other, they created a pool of following techniques:

- *Hand-n-Hold Menu*: The user controls a cursor on the screen by moving his hand in front of the screen, in a posture where all five fingers are extended. One can imagine a 2D plane right in front of the monitor which provides the coordinates of the hand position. Those coordinates always match the cursor's coordinates. The user can select an item by placing the cursor over that item for about one second.

- *Thumbs-Up Menu*: By forming a fist and position it in front of one of the displayed items, the user can highlight an item. By extending the thumb and thereby executing a thumbs up gesture one can select the highlighted item.
- *Finger-Count Menu*: All menu items are labeled with a number. By extending a corresponding number of fingers, users can select the item which is marked with that number. The used finger type is irrelevant here, just the amount of extended fingers is important here. This is a very convenient way to overcome cultural constraints, e.g. showing a two by extending thumb and index finger or showing a two by extending index and middle finger. The amount of interactable items is limited by the number of fingers, resulting in a maximum of 10 interactable items at ones.
- *3D Marking Menu*: Like the pie menu by Callahan et al. [22], which we introduced in 2.4.1, this menu is organized in a circular layout. To highlight an item, the user needs to place his fist in the center of the menu and move it towards the desired item. Moving the fist in the rough direction of the item is sufficient, to highlight it. To finally select the item, the user needs to perform a thumbs up gesture.

For each of those techniques, the user has to hold the selection pose for 0.5 seconds. So 0.5 seconds was their chosen dwell time.
Kulshreshth and LaViola conducted two experiments to measure performance and user experience. In the first experiment, they compared the Hand-n-Hold, the Thumbs-Up and the Finger-Count Menu to each other. Also, they explored, whether and how much a horizontal, a vertical and a circular layout affect the dependent variables. In the second experiment, they compared the Finger-Count Menu with the 3D Marking Menu. They were not able to compare all methods in one experiment because the 3D Marking Menu only supports the circular layout and is too different from the Hand-n-Hold and the Thumbs-Up Menu.
For all conducted experiments they measured the selection time as the time from when a random number appeared on screen to the time the corresponding number was selected. Selection accuracy was also measured as the percentage of correct selections out of total selections made. After each experiment, participants needed to fill out a questionnaire about their experiences with the technique they just used. While Hand-n-Hold was the most accurate technique, Finger-Count Menus provided by far the best selection time and the best questionnaire results. User most likely

preferred this technique because of the natural way of interaction which occasionally comes up in everyday life.
The different layouts showed no significant effect on mean selection accuracy. Horizontal layouts were significantly faster than vertical and circular layouts regarding mean selection time.
An interesting observation during the use of Finger-Count Menus was that three and four fingers were difficult to detect combination because of the higher error rate for those numbers, with three being even harder to perform than four. As mentioned before, detection is made by an optical depth camera. When showing three and four, two fingers can be too close together so that they get detected as one finger, leading to an error.
The 3D Marking Menu and Finger-Count Menu, both have the potential of being controlled without blindly or in other words, without displaying the menu at all. Kulshreshth and LaViola also tackled this topic and found out, that blindly controlling those menus will lead to a performance decrease in selection time and selection accuracy, but it is still in a well usable range.
Some users are very incompatible with the Finger-Count Menu: People who have problems with separating their fingers, e.g. people who have arthritis or many old age people who have weak intrinsic hand muscles.
Kulshreshth and LaViola are very interested in seeing the Finger-Count Menu's performance in a real application environment and also in the context of a 3D environment, e.g. in a VR environment.
We also saw an enormous potential in the Finger-Count Menu, especially because it could be possible to overcome the limits of Fitts's Law [25] and therefore we decided to fulfill their desire and make Finger-Count Menus a huge part of our work. We tested the technique against our 2D menu and our 3D physics-based menu in a VR Environment setting. To make the comparability as high as possible, we recreated the technique and the experiment as good a possible. In fact, we could recreate everything, besides the used input device, the exact same item size, item distance and item appearance. Also instead of displaying the number of the element which should be selected, we marked the element itself. During the experiment, we kept track of the same selection time and selection accuracy as Kulshreshth and LaViola. We also kept track of the numbers which are hard to interact with and which fingers were used for a successful interaction. Only because of Kulshreshth and LaViola's work, we decided to also investigate the three proposed layouts for all of our used interaction techniques.

3 Concept

In this chapter, we will present the three different menu approaches we developed and want to compare. We created a planar 2D Menu which is the simplest of the three menus. We wanted to recreate a menu similar to the one in a smartphone and thereby achieve high familiarity. The Finger-Count Menu looks the same as the 2D Menu, but to interact with it, we integrated Kulshreshth and LaViola's Finger-Count Technique [35]. And last we have the spatial 3D Menu where we wanted to make use of the available depth in VR systems and therefore we filled it with protruding elements which should increase affordance. Also, we will discuss different menu layouts and the possibilities we faced regarding submenu expansion.
For all of the following menus, we chose the same color scheme. White as a passive state, when menu items are waiting for an interaction to occur. Yellow for responsive feedback when an interaction just took place. All menu separators are cyan and all switches possess a blue and red color respectively for an on and off state. With this allocation, we wanted to enhance the experience for people with red-green visual impairments.
The whole software was written using Unity 5.5.4 [12], the Leap Motion Core Assets 4.1.6 [5], the Leap Motion Early Access Beta Interaction Engine [3] and Leap Motion's Unity Module for User Interface Input [7]

3.1 The virtual Environments

Sky Plateau

To test performance in our later user evaluation [see chapter 5], we wanted to eliminate distraction. Therefore we created a very minimalistic environment which is neutral but still immersive and to an extend visually appealing. The scene consists of a grey floor, a single grey wall and a sunset skybox. We also placed a red sphere which is used to activate a trial in the user evaluation. The single wall was placed to give the appearing menu elements a neutral and well distinguishable background 3.1.

Figure 3.1: The scene which was used during the first experiment

Apartment Room

This scene was used in our usability experiment. We wanted to create a nondistracting but still immersive environment. An environment which could be directly out of a VR game. It only had to have some lamps so that we can manipulate them with our UI, for example by turning the light on and off. We designed the scene similar to a one-room apartment. Besides the lamp, it also contains a full kitchen with some kitchenware, a refrigerator, a dresser, a couch, a shelf, a bin and a TV with its TV stand. In Figure 3.2 you can see how the whole scene looked like.

Figure 3.2: The scene which was used during the second experiment

3.2 2D User Interface (2DUI)

This is our planar menu which supports direct manipulation. It imitates a typical 2D interface which is used in tablets, smartphones or similar devices. By touching the virtual elements with the virtual hand, users can interact with the menu. While an interactable item gets touched, a sound is played as feedback and the item's color changes. The virtual hands can penetrate all parts of the menu.
For a better understanding of the 2DUI and the possible interactions, we created a short video[1] which shows the interface in action. The used prototype in the video is not the final version, but it was an iteration very close to the final version.

2D Buttons

Buttons are images with no depth at all. When users touch the buttons, they play a short sound as auditive feedback, change their color as visual feedback and activate their assigned action. Some buttons possess a continuous press feature. This means users can keep on touching these buttons to activate their assigned functionality repeatedly. Otherwise, they would need to stop touching the button and touch it again to get the same behavior. The continuous press is meant for actions which get used a lot, e.g. increasing or decreasing a given value like the body height in meter and centimeter. In contrast, the standard press should address actions which should not be pressed repeatedly, e.g. when distributing a limited amount of points, money or something similar.

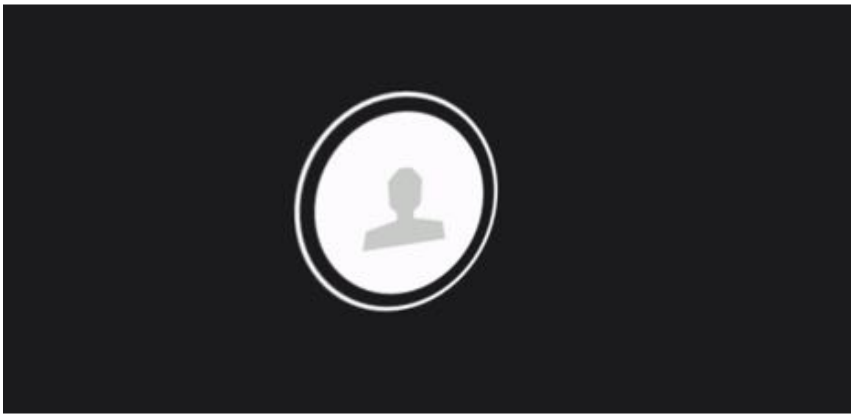


Figure 3.3: The 2D Button

[1] `https://youtu.be/Sg9WobybIvk`

2D Slider

2D Slider consist of two parts:

- A marker which is indicating the currently selected value
- A number line which suggests the available marker's range of motion

The marker slightly hovers above the number line and needs to be pressed into the line with a finger of the virtual hand, to be able to move the marker. Without pressing down the marker first, there is no possibility to move it. The marker's movement is constrained. It can only be moved along the number line. There is also a lot of auditory feedback. When the marker gets pressed down, a short clicking sound gets played. When the marker gets moved, a continuous sound gets played during the movement. On releasing the slider from the number line, another auditory feedback sound gets played.

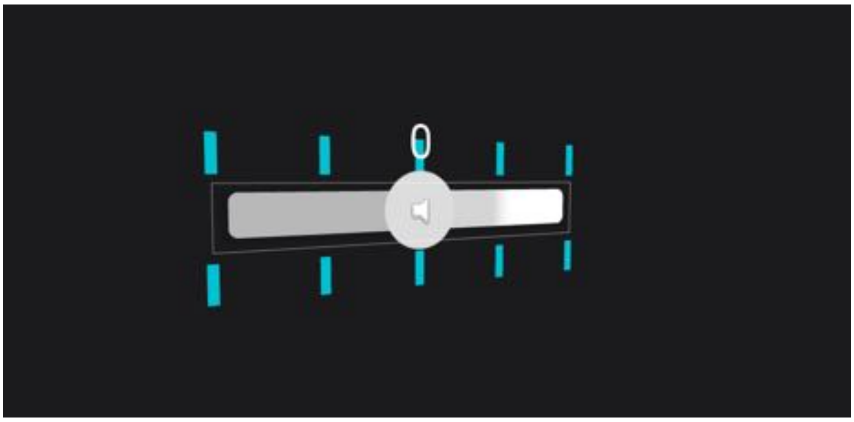


Figure 3.4: The 2D Slider

Advantages, Disadvantages and Limitations

The huge benefit of the 2DUI is the simplicity and familiarity of the interface, gained from the huge occurrence of other 2D interfaces in today's society., e.g. smartphones, tablets, desktop computers. There is not much people can do wrong, because the set of available functionality is very limited. Drawbacks of the 2DUI could be unintentionally inputs resulting out of the very small Gulf of Execution [41]. This effect could be even worse when too many UI elements are too small and too close together. The flat design of the 2DUI lacks innovation. Therefore it seems to be boring, which does not contribute to the overall user experience when used in an exiting environment like a VR game. Also, some tasks could be solved straightforward, but are very laborious to do, e.g. typing in birthday, height, etc.

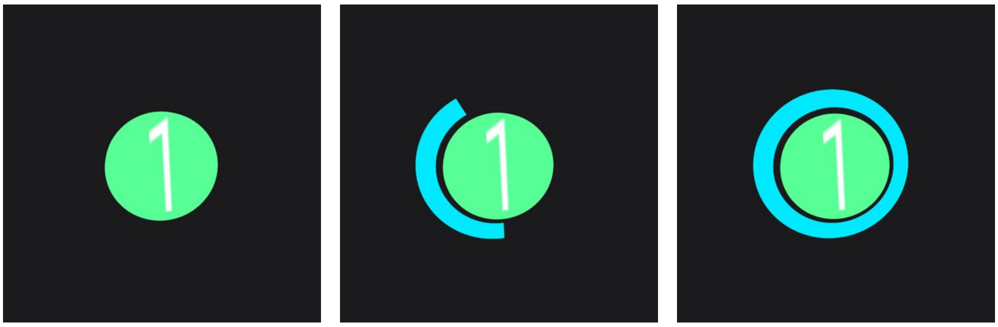
3.3 Finger Count User Interface (FCUI)

The FCUI is a new idea for menu control. It uses the number of extended fingers to control the menu. In fact, people already use this technique in everyday life, when trying to communicate a certain number to another person in a loud environment. From the appearance, it is pretty similar to the 2DUI, but all interactable elements have a label with a number beside them. To interact with the element, users need to extend the labeled amount of fingers. While using the FCUI, no element can be controlled by touching it and showing ten extended fingers will always close the current menu or submenu.
For a better understanding of the FCUI and the possible interactions, we created a short video[2] which shows the interface in action. The used prototype in the video is not the final version, but it was an iteration very close to the final version.

FC Trigger Elements

The FC Trigger Element is a green disc with a number label. When extending an equivalent amount of fingers to the number which is printed on the disc, a circular progress bar appears around the trigger element. While continuing holding up the same amount of fingers, the progress bar fills up. After a particular dwell time, the progress bar reaches its maximum and triggers another UI element which is linked to the FC Trigger Element. In our first performance experiment, we used a dwell time of 0.5 seconds. In our second usability experiment, we used a dwell time of 1.5 seconds. When changing the number of extended fingers, the progress bar resets, vanishes and needs to fill up from zero again. On interaction, the FC Trigger Element always gives auditory feedback by playing a sound and giving visual feedback by changing the color of the linked UI element.

[2] `https://youtu.be/AMZvXO2zJrI`



(a) Empty progress bar (b) Partially filled progress bar (c) Full progress bar

Figure 3.5: A FC Trigger Element

FC Buttons

The FC Buttons have the same visual representation and the same feedback as the 2D buttons. Right beside the button, there is an FC Trigger Element which is linked to the button. By interacting with the FC Trigger Element, one can also interact with the linked button. Similar to the 2D Buttons, there are single press buttons and continuous buttons. By keeping up the same number of fingers after the first interaction, a continuous button repeats the interaction in short intervals until the user changes the number of extended fingers. To interact with single press button more then once, the user needs to show a different number with his fingers and then change back to the intended number.

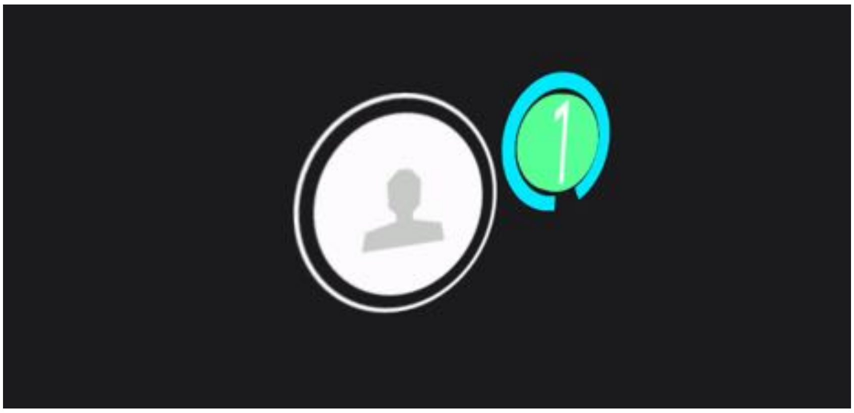
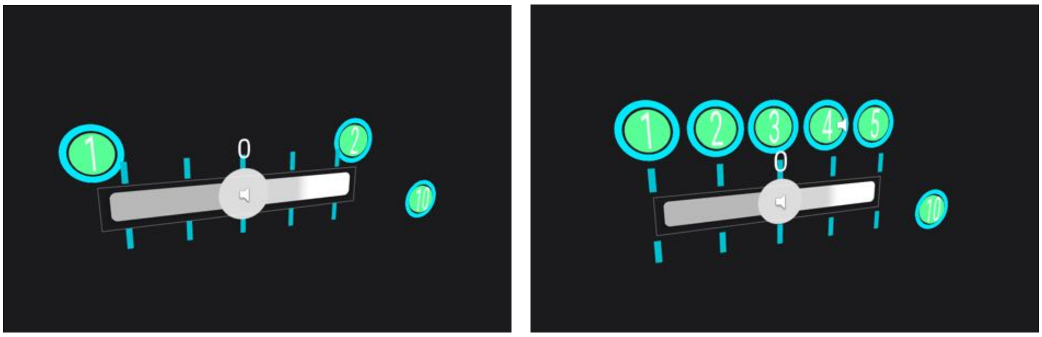


Figure 3.6: The FC Button

FC Slider

Again the visual representation matches the 2DUI, but it is equipped with FC Trigger Elements. For testing purpose, we developed two kinds of FC Sliders: A fixed slider and a dynamic slider. For the fixed slider, we can choose up to nine different positions which the slider can obtain. By extending one to nine fingers, users can move the slider's marker to one of the pre-chosen positions. The ten is needed to end the FC Slider interaction and cannot be used for positioning the marker. The dynamic slider, on the other hand, can take in any position of the slider. By showing a one or any other predefined number, the marker moves one value into the negative direction. By showing a two the marker moves one value into the positive direction. Moving the dynamic slider is a continuous interaction. As long as the user keeps on showing a one or a two, the slider keeps on moving in the corresponding direction



(a) Dynamic Slider (b) Fixed Slider

Figure 3.7: The 2 different FC Sliders

FC Digit Field

The FC Digit Field consists out of a text object which can change its currently displayed text, a background for the text object, a marker which indicates the current element and an additional FC Button. When extending an arbitrary number of fingers, the text objects will change the text to the equivalent number. After a dwell time of 1.5 seconds the shown number will be locked in and the marker switches to the next digit. During the dwell time, the same progress bar is used as for the FC Trigger Element. Every time the number, which is shown by the virtual hand, changes, the progress bar resets. Zero is an input, which needed a workaround because it is usually a state where nothing happens and users can just examine the FCUI. When showing a zero is used as an input, one would face the Midas touch problem [21] because every possible hand postures would trigger an action. Therefore we initially set the text of the text object to zero. When users want to enter a zero, they need to use the additional FC Button which will skip the current digit, and leave a zero for that particular digit. The FC Button is labeled with ten, which is the only number the text object does ignore.

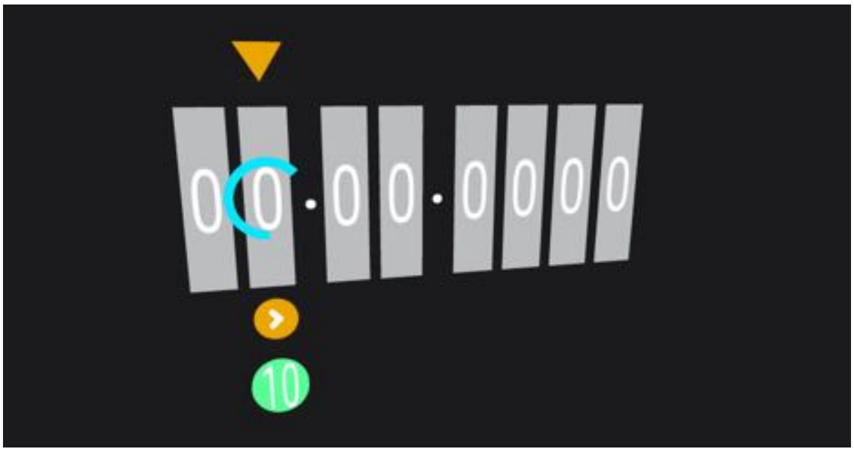


Figure 3.8: Eight FC Digit Fields besides each other

FC Picker

To choose among a set of options, users will use the FC Picker. When interacting with the picker, it displays the set of available options which are all marked with an FC Trigger Element. To chose one of the options, interacting with the trigger is sufficient.

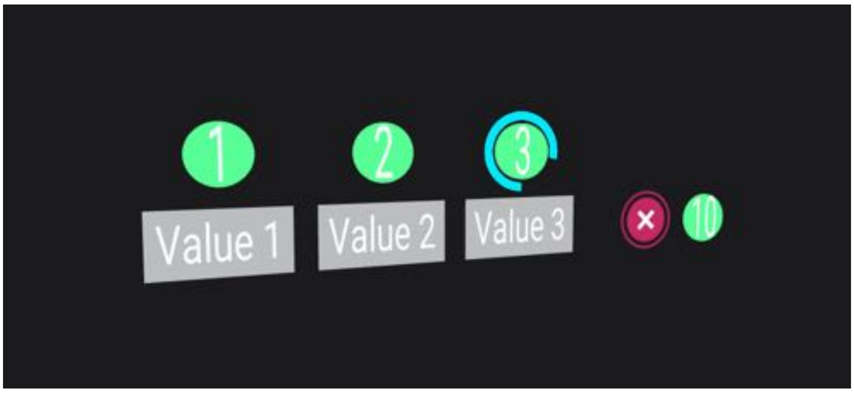


Figure 3.9: The FC Picker with three different options

Advantages, Disadvantages and Limitations

In theory, it is possible to overcome Fitts's Law [25] because aiming and moving your hand or a pointer to a specific area was omitted in this UI. Another benefit would be, that you do not need to move your whole arm for interaction, which could decrease the overall fatigue. The FCUI can be easily integrated into existing menus, by just adding FC Trigger Elements and linking them to the already existing menu items. But the greatest advantage would be the fact, which users do not need to see the interface during interaction. This fact could enable them to use a menu blindly and could also be a possible new interaction method for disabled people. As we can later see from our experiment, in practice the FCUI cannot break Fitts's Law. The performance is limited by the used dwell time. When not being careful during development, one can quickly face the Midas Touch [21] problem. Although no arm movement is needed, the high frequency of the rather uncommon hand movement is very exhausting. That is also the reason why the interface is unsuited for seniors as Kulshreshth & LaViola [35] can confirm. Logically the FCUI cannot be used by people who are missing an arm. Finally one can also say, that the FCUI is just performing as good as the used hand tracking device itself. Unprecise devices lead to hand posture errors which lead to interaction errors and finally to frustration in the usage of the FCUI.

3.4 3D User Interface (3DUI)

The 3DUI was created with the goal of maximizing affordance. All Elements are three dimensional and stick out of the menu. The form of the elements should suggest the needed actions to interact with them. We used non-isomorphic approaches to enhance their usability and feedback. The whole interaction is based on pushing the elements as if they were real physical objects. Grabbing the elements is not possible. For a better understanding of the 3DUI and the possible interactions, we created a short video[3] which shows the interface in action. The used prototype in the video is not the final version, but it was an iteration very close to the final version.

3D Buttons

3D Buttons consist out of a fixed in space trigger plane and a movable disc right in front of the plane. When pushing the disc into the plane, an interaction gets

[3] `https://youtu.be/fyDA0pgevH4`

triggered and plays an auditory feedback. The plane becomes transparent when the movable disc gets pushed through it. Buttons can possess the continuous press feature, which enables repeated interaction while continuing touching the disc behind the plane. Single press buttons need to be pressed through the plane multiple times to bring forth repetitious interactions. While the disc is not on its original position, it strives to reattain the position.

(a) From the front

(b) From the side

Figure 3.10: The 3D Button

3D Slider

Like the 2D Slider and FC Slider, the 3D Slider consists out of a marker and a number line. By pushing the marker like a real physical object, you can move the slider with the virtual hand. During the movement, you can hear a continuous sliding sound.

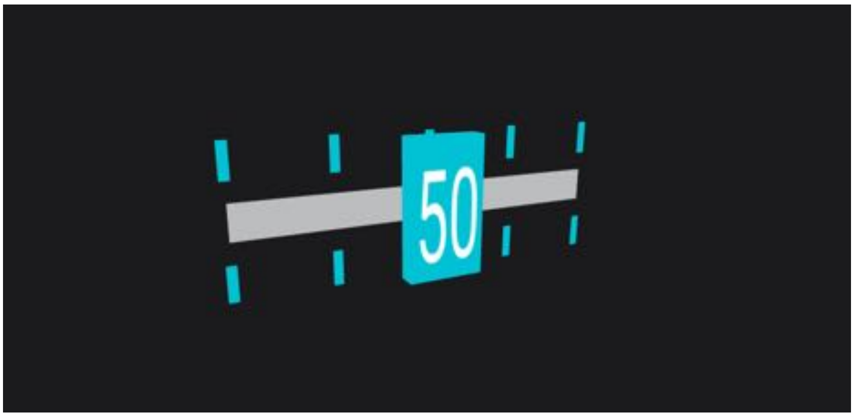


Figure 3.11: The 3D Slider

3D Wheel Picker

The 3D Wheel Picker is a decagon shaped wheel. Each of the wheel's ten faces contains a text object with a replaceable text. There is a fixed location where faces get marked as selected and thereby change their color. At any time there is always exactly one marked face. By pushing the wheel like a real physical object, users can apply force, rotate the wheel and thereby select a different face. Every time the currently selected face switches, a short clicking sound gets played. It is also possible to use more than ten texts and simulate an infinite wheel. This infinite wheel is the perfect example for a combination of an isomorphic and nonisomorphic approach [see chapter 2.2].

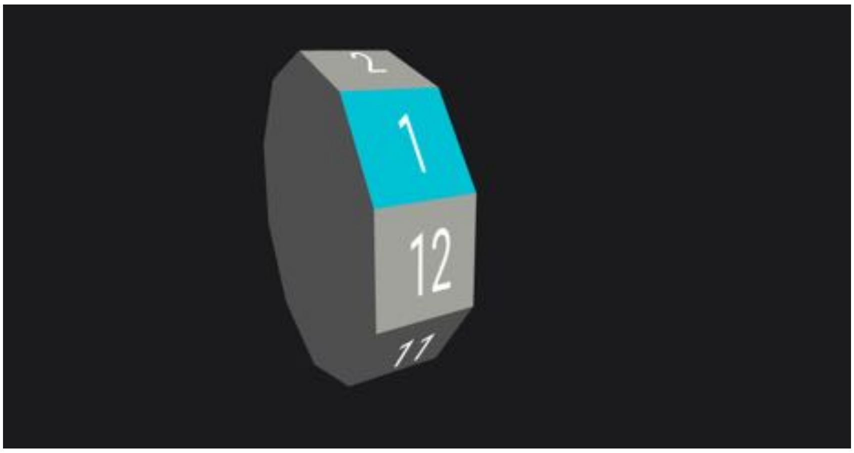


Figure 3.12: The 3D Wheel Picker

3D Switches

During the development, we came up with three different types of switches.

3D Toggle Switch

Toggle switches are comparable to the switches which are used in plane cockpits. By flipping the switch from one side over its center, it automatically moves to the end of the opposite side and triggers an action. Upon crossing the center, one of two short sound gets played, depending on whether the user turns the slider on or off.

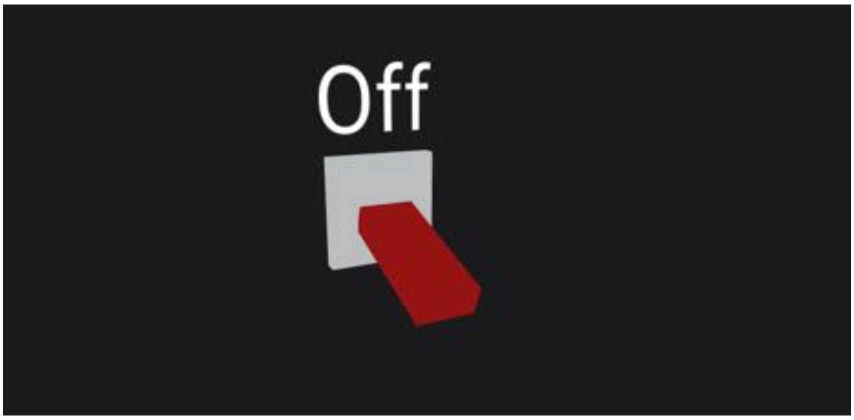


Figure 3.13: The 3D Toggle Switch

3D Rocker Switch

Rocker switches are comparable to conventional light switches. After pushing it over the center, the element behaves just like the 3D Toggle Switch. It automatically moves to the end of the opposite side, triggers an action and plays a short sound.

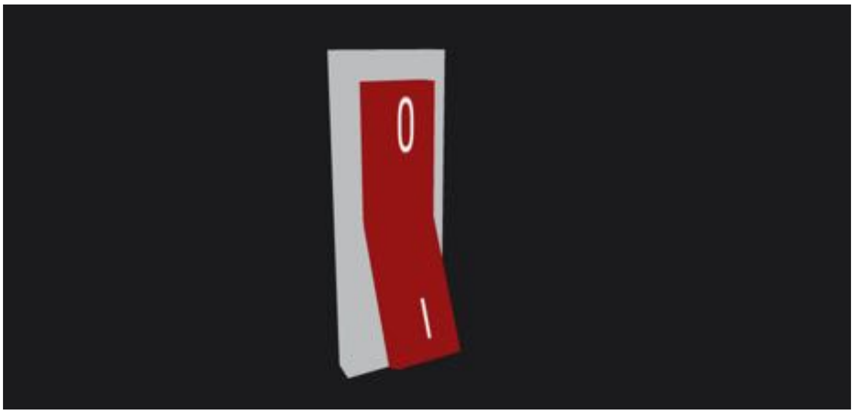


Figure 3.14: The 3D Rocker Switch

3D Slider Switch

This slider was inspired by the typical Slider Switches in Android or iOS phones. We shortened the number line of the 3D Slider and combined with the flipping mechanism of the 3D Toggle Switch and the 3D Rocker Switch. So when the slider switch is halfway on the other side, it automatically travels the remaining distance to the end.

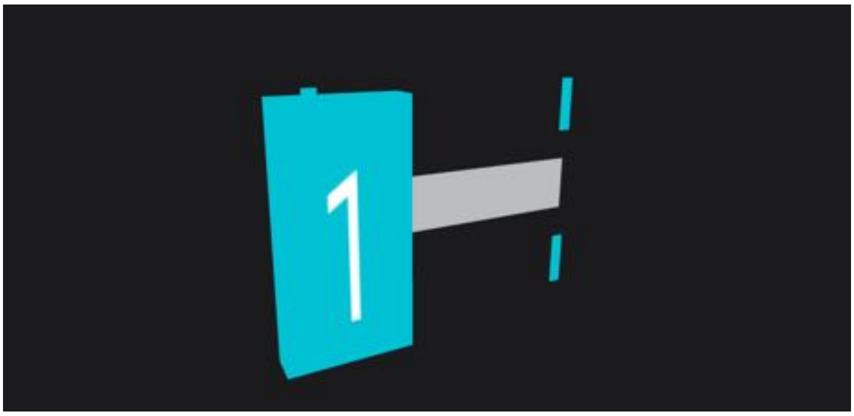


Figure 3.15: The 3D Slider Switch

Advantages, Disadvantages and Limitations

The 3DUI is very engaging, interactive and is fun to use. It utilizes pseudo feedback through a combination of an isomorphic and nonisomorphic approach, which tries to compensate the lack of haptic feedback. Because the 3DUI uses a lot of real-world metaphors and analogies, its behavior can be easily predicted by users. But still, the 3DUI suffers from the lack of haptic feedback because pushing an object without feeling anything on the hand seems to be odd.

3.5 Menu Structure and Widgets

In order to analyze usability we first needed to create a realistic menu environment. To create such menus, we used all of the mentioned elements and combined them into small widgets. For each interface representation (2D, 3D and Finger Count) we created a separate menu which consists of the related elements only.

Menu Navigation Buttons

We used a main menu as a starting point of the whole menu. By using one of the four button elements, users will be lead to a submenu. There is also a button which can close the main menu. So all buttons which serve the purpose of navigating from one submenu to another, are classified as Menu Navigation Buttons.
The four submenus are structured as follows:

1. Me Menu: The first layer of the Me Menu is a navigation menu just like the main menu. You can either access the Personal Info Menu or the Skills Menu through it.
    a) Personal Information Menu: In this submenu, users can set personal data about themselves or a fictional character.
    b) Skills Menu: As the name suggests, users can distribute points for different attributes in the virtual environment.
2. Items Menu: After entering the Items Menu, users can choose between 4 different buttons, which lead to 4 different items.
3. Environment Menu: The environment menu can manipulate the state of the virtual environment.
4. Settings Menu: In this menu, it is possible to alter systems generic system settings.

(a) 2D Representation (b) 3D Representation (c) FC Representation

Figure 3.16: The Main Menu

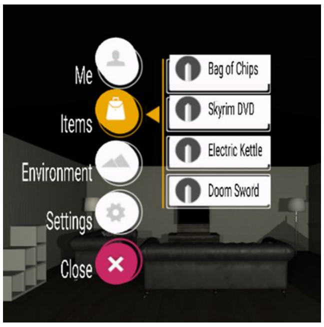
(a) 2D Representation

(b) 3D Representation

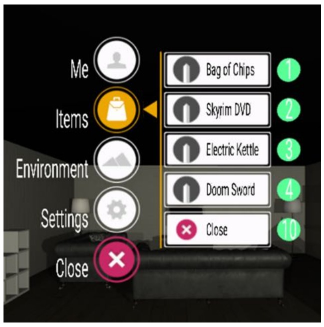
(c) FC Representation

Figure 3.17: The Me Menu

(a) 2D Representation



(b) 3D Representation



(c) FC Representation

Figure 3.18: The Items Menu

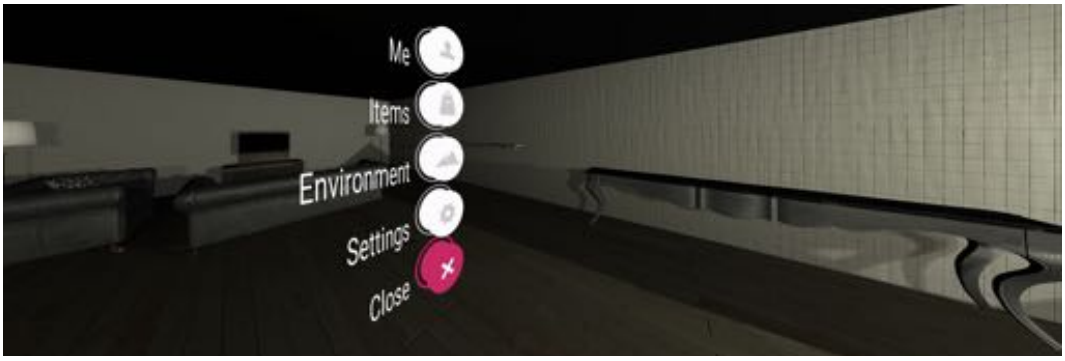


Figure 3.19: The 3D Main Menu from the Side

Name Input Widget

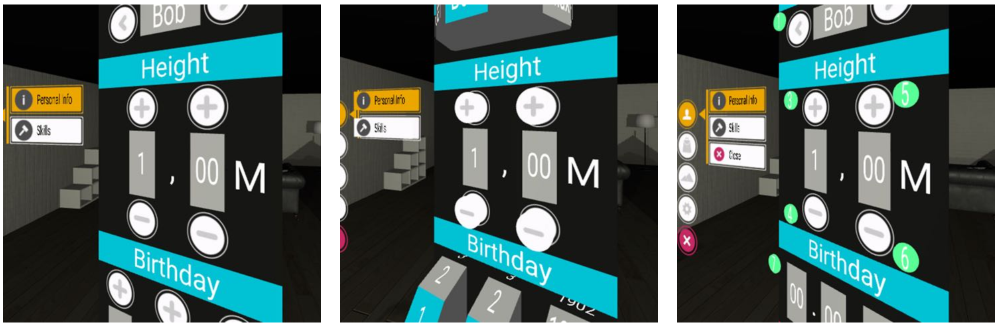
Entering the name is a task which belongs to the Personal Information Menu. VR text input is a huge and still researched topic, but not the main focus of this work. So we did not provide an option to input a custom text. Instead, we gave the users a set of names from which they could choose one. This widget uses two 2D Buttons in the 2DUI, a labeled 3D Wheel Picker in the 3DUI and two FC Buttons for the FCUI. The used buttons where single press buttons because otherwise, users would scroll through the names too fast, which would negatively affect usability.

(a) 2D Representation (b) 3D Representation (c) FC Representation

Figure 3.20: The Name Input Widget

Height Input Widget

This task also concerns information about the user and therefore it is also located in the Personal Information Menu. The 2DUI uses four 2D Buttons in total to increase or decrease the meter value or the centimeter value of the height by one. The 3DUI uses four 3D Buttons and the FCUI uses four FC Buttons to solve the same task. Because there are a lot of possible values to choose from, the mentioned buttons were continuous buttons.



(a) 2D Representation (b) 3D Representation (c) FC Representation

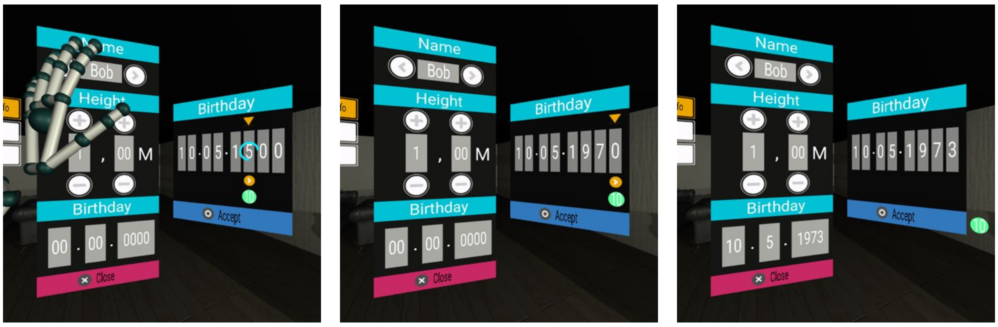
Figure 3.21: The Height Input Widget

Birthday Input Widget

The last widget of the Personal Information Menu concerns the input of a date in the european format. To enter a date, users need to enter three numbers with eight digits. So we used six 2D Buttons in the 2DUI to handle this problem. A pair of two 2D Buttons was used to alter the values of the day, month and year, respectively. Again those buttons were continuous buttons. The 3DUI uses three infinite 3D Wheel Picker to set those three values. For both the 2DUI and the 3DUI it is only possible to set a value between one and 31 as the day, a value between one and twelve as the month and a value between 1900 and 2017 as the year. Finally, the FCUI utilizes 8 FC Digit Fields to enter both digits of the day and the month and all four digits of the year. We used the european date dormat, so they had to enters the day first, then the month and lastly the year. It was possible to enter every number from 0 to 9 into each FC Digit Field.

(a) 2D Representation (b) 3D Representation (c) FC Representation

Figure 3.22: The Birthday Input Widget

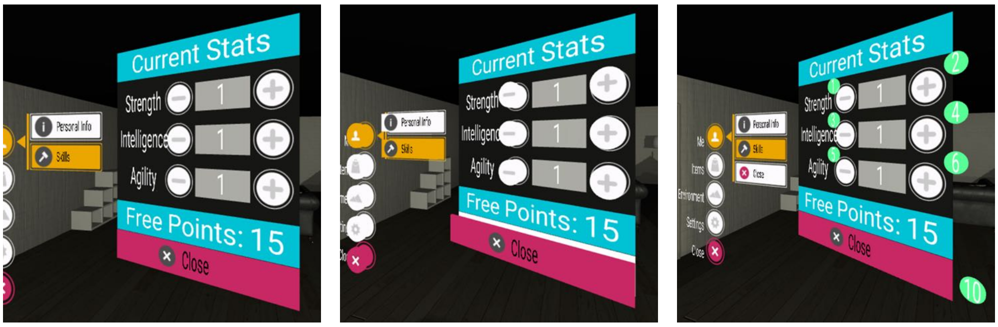


(a) Show Number (b) Repeat until End (c) Accept by showing ten Fingers

Figure 3.23: Input process of the FCUI

Skill Distribution Widget

We used three Skill Distribution Widgets in the Skills Menu. Each of those three Widgets uses two buttons of the corresponding UI (2D, 3D and Finger Count). With those buttons, users can increase or decrease the value of a certain attribute. All buttons belong to the single press button type. This decision should serve as a safety mechanism to prevent unintentional interaction because the amount of increase is limited by the number of free distribution points in the skills menu.



(a) 2D Representation (b) 3D Representation (c) FC Representation

Figure 3.24: The Skill Distribution Widget

Item Viewer Widget

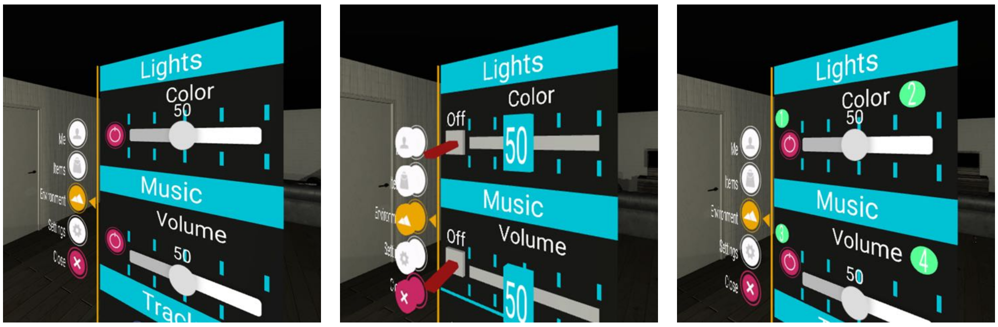
The Item Viewer Widget gets used in the Items Menu to display and interact with the selected items. A 3D model of the item floats above a grey disc. By pushing the model with the virtual hands, one can rotate the item around its center. The widget also contains a board with a description of the item. At this point it is important to mention that the small grey disc serves the purpose of helping the user to differentiate between a menu item and an environment item. Without the disc, it is possible that users get confused there.

(a) 2D Representation (b) 3D Representation (c) FC Representation

Figure 3.25: The Item Viewer Widget

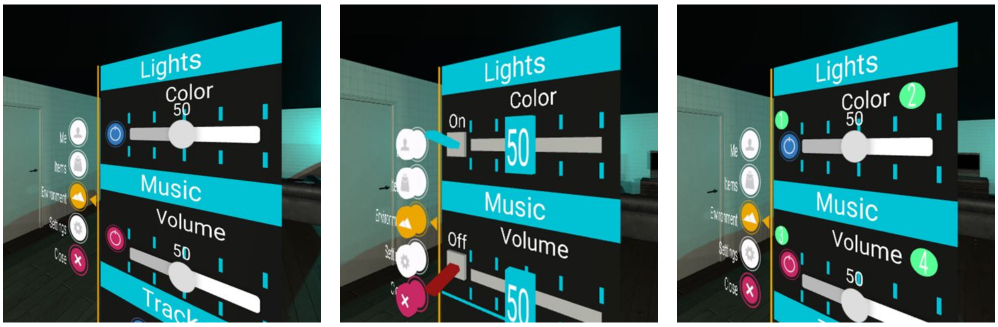
Light and Music Switches

We included some light and audio sources into the virtual environment. To control those we build in some control elements into the Environment Menu. The most basic action to perform on light and music would be turning them on and off. To fulfill this task, we use 2D Buttons in the 2DUI, 3D Toggle Switches in the 3DUI and FC Buttons in the FCUI. Interacting with them will turn on or turn off the light and change the visual representation of the button into an on or off state.



(a) 2D Representation (b) 3D Representation (c) FC Representation

Figure 3.26: The Light and Music Switches in an OFF state

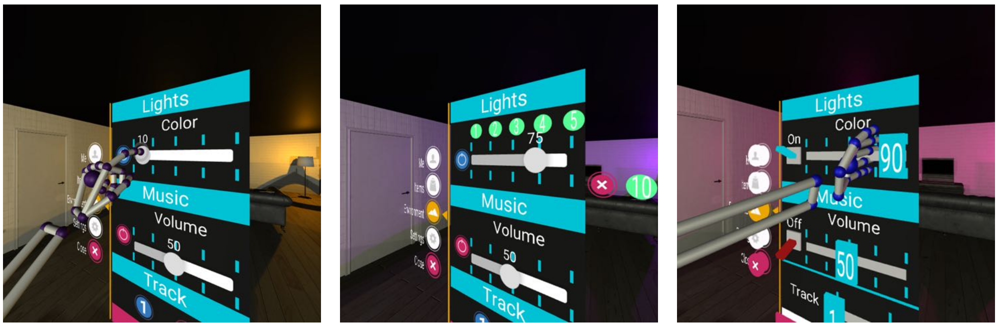


(a) 2D Representation (b) 3D Representation (c) FC Representation

Figure 3.27: The Light Switch in an ON state

Hue and Volume Slider

A slightly advanced task concerning light and music would be manipulating the light's color and the music's volume. Therefore the Environment Menu also contains two 2D Sliders in the 2DUI, two 3D Sliders in the 3DUI and two fixed FC Sliders in the FCUI. The sliders are located right beside the Light and Music Switches.



(a) 2D Representation (b) 3D Representation (c) FC Representation

Figure 3.28: The Hue Slider with a changed value

Track Switcher

The last task we tackled in the Environment Menu is changing between two possible music tracks. In the 2DUI and FCUI we used two 2D Buttons and two FC Buttons, respectively, which were labeled with the track number they will activate. When a track is already playing and the button to switch to the same is pressed again, then nothing happens. In the 3DUI we used a 3D Slider Switch, which we specially created for this task. We even integrated an interactable transition effect into the 3D Slider Switch depending on its marker position. When the marker is in the center of the switch, the music gets muted. When it remains at one of the two boundaries, then the music gets played with 100% of the currently set volume. So when sliding the marker from left to right or the other way round, it will create a fading effect. The volume of the currently played track will decrease until the marker hits the center. At the center, the track will switch, and on the marker's way to the edge of the switch, the volume will increase again.

(a) 2D Representation (b) 3D Representation (c) FC Representation

Figure 3.29: The Track Switcher

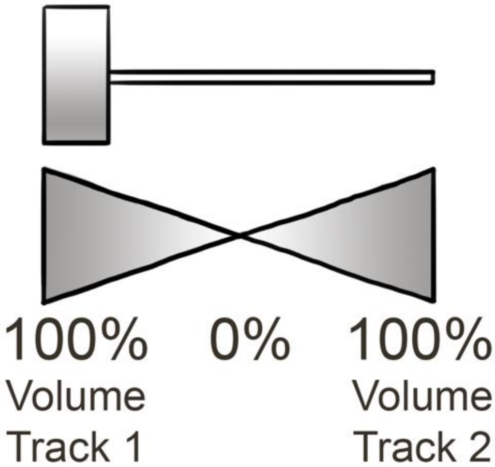
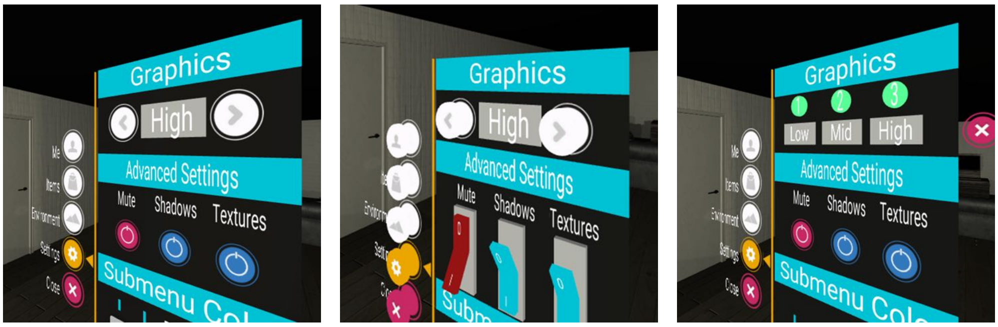


Figure 3.30: The Fading Effect of the 3D Slider Switch [32]

Graphics Quality Switcher

The Graphics Quality Switcher is the first widget of the Settings Menu. In many applications which need to render a 3D scene, it is well-known use to provide an option to switch the graphics quality depending on the power of the graphics card. We offer the option to set the graphic to low, mid and high, but we are not actually changing the render quality. For our testing purpose, we just replaced the 3D models of the apartment room [see chapter 3.1] with ones that have more or fewer details. To switch between the provided options, the 2DUI uses two 2D Buttons and the 3DUI uses two 3D Buttons. So for those two UIs, the setup is pretty similar to the

Skill Distribution Widget. The FCUI uses one FC Picker which enables users to make their choice directly.



(a) 2D Representation (b) 3D Representation (c) FC Representation

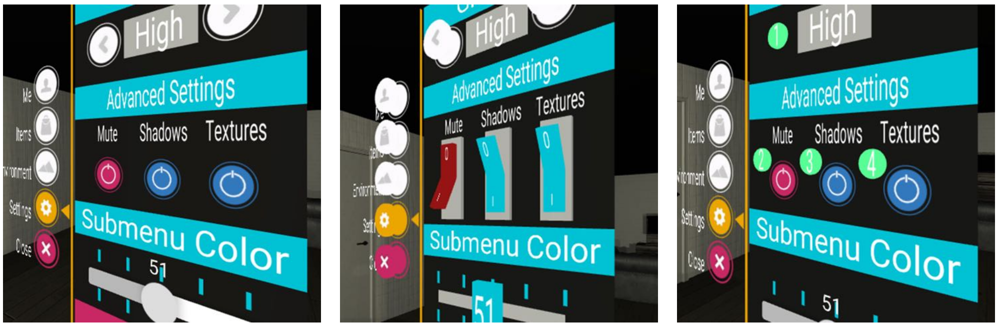
Figure 3.31: The Graphics Quality Switcher

(a) Low Quality (b) Mid Quality (c) High Quality

Figure 3.32: The Apartment with different Graphics Quality Settings

Advanced Settings Switches

The second widget of the Setting Menu is responsible for turning sound, textures and shadows either on or off. In the 2DUI we use three 2D Buttons and in the FCUI we use three FC Buttons to fulfill this task. The 3DUI makes use of three 3D Rocker Switches instead.



(a) 2D Representation (b) 3D Representation (c) FC Representation

Figure 3.33: The Advanced Settings Switches

Menu Color Slider

The Menu Color Slider is the last missing widget of the Settings Menu and also of the whole menu environment. For a better overview, we partitioned all the submenus by their tasks. Those partitions were achieved by using headings between the different widgets. With this slider, it was possible to change the background color of those headings. The 2DUI and 3DUI uses a 2D Slider and a 3D Slider, respectively. The FCUI uses a dynamic FC Slider to precisely set those values.

(a) 2D Representation (b) 3D Representation (c) FC Representation

Figure 3.34: Menu Color Slider

4 Implementation

4.1 2D User Interface (2DUI)

2D Buttons

The 2D buttons are an image plane with a hitbox around them. The hitbox is slightly larger than the visual representation of the button. As soon as the virtual hand's collider enters this hitbox, an interaction occurs which triggers an action. Buttons can be single press buttons or continuous buttons. To interact more than once with a single press button, the user's hand needs to leave the hitbox and reenter it. With a continuous button, the interaction will repeat in short intervals as long as the hand remains in the hitbox. Whenever an interaction takes place, a short sound will be played as auditory feedback and the button's color changes from white to yellow as visual feedback.

2D Slider

The 2D Slider was provided by the Leap Motion UI Input Toolkit [7]. All sounds came from the same library. We just added a little text object which displays the currently chosen value.

4.2 Finger Count User Interface (FCUI)

FC Trigger Elements

We outsourced the code which detects the number of extended fingers and handles the interaction with the appropriate trigger element into an own controller object. To figure out the number of extended fingers, we used Leap Motion's Frame Class [6] contains all available information about the Leap Motion Hand in the last rendered frame. The controller also knows all currently displayed FC Trigger Elements and when the FC Trigger Elements change. This is important during menu navigation because otherwise, unintentional behavior would occur. For example, when showing a two to enter a submenu, the old trigger element which opened the submenu will disappear and potentially a new trigger element which is also labeled with a two could appear. If the controller would not know, that the trigger elements were replaced, a user would automatically start filling the progress bar of the new trigger element. When the dwell time is too short to react, the user would then trigger an unintended action. The controller also manages the timer and just informs the respective FC Trigger Element, that it should execute its linked functionality. The linked functions are easily exchangeable because we used the unity event system [9] for it.

FC Buttons

FC Buttons are 2D Buttons where we just removed the collider/hitbox so that users can not interact with it by touching the button. We used the FC Trigger Elements link system to trigger the internal 2D Button functions.

FC Slider

Those sliders are just basic 2D sliders with additional FC Trigger Element. When an interaction with one of the fixed slider's triggers occurs, the slider's setValue function is called, which automatically sets the value and moves the marker to the corresponding position. The dynamic slider also uses the setValue function and decreases or increased the current value by a small amount. In our case, we had a number scale from zero to 100 and altered the value by one whenever an interaction occurs. Combined with the continuous interaction, users can cover a huge scale of values rather quickly.

FC Digit Field

The text object is always grabbing the number of currently extended fingers from the separate controller mentioned in the FC Trigger Elements implementation [see chapter 4.2] and sets its text to the same number. The current timer's state and thereby the progress bar are also grabbed from the controller. The FC Digit Field itself is just handling the movement of the marker when users need to input several digits.

FC Picker

The picker just opens a new submenu which contains several text objects which are linked to the same amount of FC Trigger Elements plus one additional trigger for canceling the selection and quitting the submenu again.

4.3 3D User Interface (3DUI)

3D Buttons

Both the trigger plane and the disc possess a hitbox. When those two hitboxes collide, an interaction occurs and an internal boolean gets set. The internal causes to make the plane transparent and states, that the button is behind the trigger plane. When the disc gets moved back in front of the plane, the boolean switches its state. The force which is applied to move the disc back to its original position is caused by a unity component called spring joint [11]. Just like a real spring, the further the disc is away from its initial state the more force is applied to restore it. The disc itself is movable by the virtual hand because it is a rigid body.

3D Slider

The number line serves no functionality. It is only a visual aid for the user, to show him the different areas of the slider and its boundaries. The marker is a rigid body and has its movement regulated by a configurable joint [8]. By calculating the distance between the boundaries which are set by the configurable joint and comparing it to the current position of the marker, we can calculate the currently selected value of the slider.

3D Wheel Picker

The wheel itself is a rigid body. Additionally, every face has an own hitbox. The area where the currently selected object switches is a hitbox itself. When a new face hitbox enters the selection hitbox, the last entered face will be deselected and the new hitbox gets selected. The wheel's movement is defined by a configurable joint [8] which is fixed in position and can only rotate along one axis. When simulating an infinite wheel, we use two hitboxes similar to the selection hitbox, which replace a placeholder's text on the backside of the wheel [see Figure 4.1]. For the user who is looking at the wheel from the front, it seems like there are unlimited texts on it. When users wanted to switch between two faces which are close together, it was a pretty hard task at first. We implemented a function which increased the wheel's drag during low velocity and decreased the drag during high velocity. This lead to a far better performance and usability.

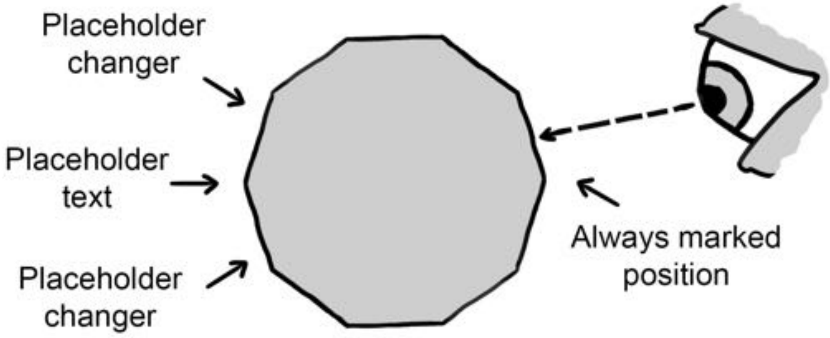


Figure 4.1: The Concept of the Infinite 3D Wheel Picker [32]

3D Switches

3D Toggle Switch and 3D Rocker Switch

The toggle switch and the rocker switch, both consist out of 2 parts. A base, which is just a visual aid for the metaphor and a small lever or rocker which can be moved with a single finger. The movement was solved using unity's hinge joint [10]. When passing a specific angle, we just inverted the force which is constantly applied to the lever by the joint. At that crucial moment, we also trigger all linked actions and switch the switches color.

3D Slider Switch

Similar to the 3D Slider we used a configurable joint [8] to handle the movement. But in contrast to the 3D slider, we constantly apply force on the marker which is thereby held in place at one of the two boundaries of the switch. When the marker reaches half the distance between the two boundaries, we just invert the force of the joint again and trigger the linked actions.

5 User Evaluation

5.1 Participants

We recruited 11 participants (7 males and 4 females ranging in age from 21 to 31) from the University of Saarland, of which all were right-handed. 3 of them are visually impaired wore glasses and 8 of them had prior VR experience. The experiment duration ranged from 70 to 90 minutes and all participants were allowed to enjoy a VR experience of their choice afterwards (15 minutes).

5.2 Apparatus

The experiment's setup consisted of one HTC Vive which was mounted and lead through the ceiling, a Leap Motion Controller which was attached to the HTC Vive to be able to track the hand motion while the participant is still able to move freely [see 5.1]. The Leap Motion was connected directly to the HTC Vive and transmitted the data through its USB hub to the PC. The used Computer had an i7-5820K, a GTX 980 and 16GB DDR4 Ram build in. This set-up offered far more than enough computational power, to conduct the study without noticeable frame drops (below 90 FPS for longer than 0,5s).

Figure 5.1: The used Output (HTC VIVE) and Input (Leap Motion) Setup [1]

The Unity3D game engine (Version 5.5.4) [12] and the Leap Motion SDK for Unity (Core Assets 4.1.6) [5] was used for implementing all interface elements. Especially the Leap Motion Interaction Engine Early Access Beta [3] was a fundamental part of this work.

5.3 Hypothesis

H_1 The 3DUI will be preferred by the users regarding usability and user experience.
H_2 The 2DUI will achieve the best performance results regarding speed and accuracy
H_3 In VR, there is no significant difference in performance (speed and accuracy) regarding different menu layout

5.4 Design

To measure and compare user performance and preferences regarding the three different user interface approaches, we decided to conduct two separate experiments.

[1]Image by pumpkinbundtcake with his approval of use `http://i.imgur.com/PcOPUWM.jpg`

Comparing the 3 different approaches to each other is very important here, so for both experiments the within-subjects design was chosen.

5.5 Experiment 1: Performance

For the first experiment, we tried to achieve the highest comparability possible to Arun Kulshreshth's Finger-Based 3D Gesture Menu Selection [35]. The experiment concerns performance and has 3 independent variables:

- Menu appearance (Plane 2D, Spatial 3D)
- Interaction Mode (Direct Manipulation, Finger-Count)
- Layout (horizontal, vertical and circular)

This would make 2x2x3=18 conditions, but the spatial 3D menu is very affordance intensive and would by design rather suggest direct manipulation instead of Finger-Count Inputs. So we decided to not combine the Finger-Count interaction mode with the Spatial 3D Menu. Like this, we have 3 Interfaces: Plane 2D with Direct Manipulation, Spatial 3D with Direct Manipulation and Plane 2D with Finger-Count. In total, this leads to 3x3=9 conditions. The user conducted 10 trials for each condition which results in 90 interactions per participant. Each trial ends if the participant selects the right target with the given interaction mode. The dependent variables were the same as Kulshreshth and LaViola's variables. This means the average menu selection time and selection accuracy results from the average values over the conducted 10 trials for a certain condition.
To ensure that every trial has a comparable starting position, we made sure that every experimental subject has to hold their real hand and the virtual hand at a marked location in virtual space. Therefore we used an initially red sphere as a marker. When touching the sphere with the virtual hand, it becomes purple. Only when the participants have no fingers extended or in other words, as soon as the form a fist, the sphere begins to transition from a purple color to a green one. The transition will be interrupted immediately if the hand does not intersect with the sphere at all or if the participants extend any number of fingers. After 1 second the transition is completed, the sphere vanishes and either a new trial begins or the condition switches.
The red and green color of the sphere is problematic for people who suffer from dyschromatopsia, but we could not change the colors or the type of transition. Yellow

and blue are already used for the targets and could decrease visibility if the sphere has the same colors and is right in front of them. Also, it could lead to confusion for the participants. The transition could not be changed because either the participants have to look away from the targets or the alternative transitions we came up with were very distracting. A simple color lerp was the most subtle option. Therefore we excluded everybody who is affected by dyschromatopsia.
A pilot study showed that it is very confusing for the subject to adapt from one condition to another if the interaction mode switched. To decrease the possibility of frustration caused by the study design and not by the interfaces themselves, we decided to group the 9 conditions by the 3 interfaces. This means the Interface only switches after conducting the trials with every of the 3 layouts (horizontal, vertical, circle).
After each condition switch the subject has some time to get familiar with the new condition. With the subject's confirmation of being ready, the conductor starts the new task and the sphere appears.
For every trial, one of five possible targets, which are numbered from 1 to 5, will be marked with a blue color. If the participants select any of the targets with the given method, it will give feedback by turning yellow and playing a sound. As mentioned before: The trial ends with the selection of the marked target and the sphere appears again. A logger was implemented to keep track of following dependent variables:

- Completion Time
- Selection Accuracy (Interactions with elements which were not marked)

The logger also kept track of the amount and the kind of extended fingers, which were used to interact with the marked element successfully.
For the first experiment it is also important to mention that before entering any test condition, the participant will get some time to learn how to interact with the sphere. Also, this first experiment was designed and should only be conducted by using one hand only. The used hand should be the experimental subject's dominant hand. It is not allowed to switch hands during the experiment, otherwise it could influence the consistency of the arm fatigue variable in the following questionnaire.
To address counterbalancing, a full counterbalance was used for the 3 interfaces resulting in 6 different permutations. For counterbalancing the layouts, we randomized the sequence of layouts similar to the Latin Square design, resulting in following ordering:

Horizontal, Vertical, Circular	Vertical, Circular, Horizontal	Circular, Horizontal, Vertical
2D + Direct Manipulation	3D + Direct Manipulation	2D + Finger Count
2D + Direct Manipulation	2D + Finger Count	3D + Direct Manipulation
3D + Direct Manipulation	2D + Direct Manipulation	2D + Finger Count
3D + Direct Manipulation	2D + Finger Count	2D + Direct Manipulation
2D + Finger Count	3D + Direct Manipulation	2D + Direct Manipulation
2D + Finger Count	2D + Direct Manipulation	3D + Direct Manipulation

Table 5.1: The Ordering of the Layouts and Interfaces

The order of the marked targets is randomly generated but still evenly spread beforehand. This means every target will be marked twice for every condition and every experimental subject will get the same sequence of target numbers.

Post Experiment 1 Questions	
Q1	To what extent did you like this menu selection technique?
Q2	How mentally demanding was this technique?
Q3	To what extent your arm was tired when using this technique?
Q4	Did you feel hurried or rushed when using this technique?
Q5	How successfully you were able to choose the items you were asked to select?
Q6	Did you feel that you were trying your best?
Q7	To what extent you felt frustrated using this technique?
Q8	To what extent did you feel that this technique was hard to use?
Q9	Which layout of menu items would you prefer for this technique? Horizontal, vertical, circular or all equally?

Table 5.2: Questionnaire used at the end of the first Experiment

At the end of the first experiment, the participants should fill out a questionnaire [see Table 5.2]. For each Interface the same set of questions was used and measured following dependent variables:

- Q1: Overall Best
- Q2: Mental Demand
- Q3: Arm Fatigue

- Q4: Pace of the Interface
- Q5: Selection Rate
- Q6: Subject's Effort
- Q7: Frustration
- Q8: Difficulty
- Q9: Layout Rating

In order of being able to make a better comparison between interfaces, if they achieve a similar rating, the participant should also assign ranks to every Interface. Rank 1 would be the interface which they liked the most and rank 3 would be the interface which they liked the least.

5.6 Experiment 2: Usability

For the second experiment, we wanted to further investigate in the UX Design. The experiment concerns usability and has two independent variables with following levels:

- Menu appearance (Plane 2D, Spatial 3D)
- Interaction Mode (Direct Manipulation, Finger-Count)

This makes 2x2=4 conditions, but like in experiment 1, the spatial 3D menu is very affordance intensive and would by design rather suggest direct manipulation instead of Finger-Count Inputs. So again, we did not combine the Finger-Count interaction mode with the Spatial 3D Menu. Like this, we have the same 3 Interfaces again: Plane 2D with Direct Manipulation(2DUI), Spatial 3D with Direct Manipulation(3DUI) and Plane 2D with Finger-Count(FCUI).
Like this, we have three conditions in total. The participants should complete 26 different tasks per Interface [See Table 5.3].

Task ID	Task description
1	Go to the Personal Info Menu
2	Choose as your name: ANA. It is written with just one N
3	Choose as height: 2 Meter 25cm
4	Choose as your Birthday: 23.04.1987 in the European format. This means the 23rd as the day at the first position. The Fourth/April as the month at the second place and 1987 as the year at the last position
5	Go back to the main menu
6	Go to the Skills Menu
7	Set the points of an attribute (Strength, Intelligence or Agility) to a total of 6
8	Go back to the main menu
9	Go to the Items menu and read out loud, the hidden Code on one of the items (Bag of Chips, Skyrim DVD or Doom Sword)
10	Go back to the main menu
11	Go to the Environment Menu
12	Turn on the lights
13	Set the light's color to purple, by moving the slider to a value of exactly 75
14	Turn on the music
15	Lower the music volume to a value of exactly 25
16	Switch to Track 2
17	Switch to Track 1
18	Turn off the music
19	Please go back to the main menu
20	Please go to the Settings Menu
21	Set the Graphics Quality from "High" to "Low
22	Turn off the Textures
23	Turn off Shadows
24	Set the Submenu Color to Orange, by moving the slider to a value of exactly 10
25	Please go back to the main menu
26	Close the whole Interface

Table 5.3: Tasks which should be fulfilled in the second Experiment

Each task ends when the participant communicates to the conductor, that he or she thinks that the task was solved and is finished. The observed dependent variables were:

- Solving Time
- Selection Accuracy
- Success Rate
- Evaluation of the Questionnaires.

Time, accuracy and success tracking will be handled automatically by the system. The conductor just needs to mark, when the task started.
Before letting the participants face the task with an interface, they get some time to get familiar with it. In contrast to the first experiment, the users are allowed and partially need to use both hands.
We addressed counterbalancing for the interfaces by a full counterbalance, resulting in 6 different interface permutations.
The 26 tasks were also counterbalanced. They were grouped depending on the submenu they were in. This decision makes the experiment less time consuming because a participant would not need to navigate from one submenu to another for every task. All interfaces have the same 5 submenus:

- Personal Information
- Skills
- Items
- Environment
- Settings

The order of the submenu occurrence was counterbalanced by a 5x5 balanced latin square resulting in ten different submenu occurrence permutations.
The Personal Information menu tasks were counterbalanced by a full counterbalanced of three tasks, resulting in six different permutations. For every interface iteration, another permutation is picked again, to reduce learning effects between interfaces.
The Skill and Item menu tasks are both counterbalanced by a full counterbalance of three tasks, resulting in six different permutations. Every participant gets another

permutation and every interface iteration only one of the three tasks will be done. The tasks are basically the same and the only purpose of having different tasks in those two menus, is to prevent the participant from getting bored by doing nearly the same task in every iteration again. The interface does not change significant enough for those tasks, to classify them as an entirely new experience.
The Environment Menu consist out of two task blocks. Task blocks are a sequence of tasks which need to be executed in a certain order. Otherwise, some tasks would not make any sense at all, e.g. lowering the volume of music which is not even playing, yet. Those two task blocks were counterbalanced by a full counterbalance, resulting in two different permutations. For every interface iteration and every participant the permutation changes.
The five Settings menu tasks are counterbalanced by a balanced latin square, resulting in ten different task permutations. Every interface iteration and every participant a different permutation is picked.
Because of the ten submenu and settings permutations we needed at least ten participants, to test every order at least once.
At the end of every interface iteration, the participants should fill out a questionnaire concerning Motion Sickness, Immersion, User Experience and a NASA TLX.
At the end of the experiment, the participants should give their rating to every interface element individually and rank them from Rank 1 (best) to 3 (worst) regarding overall favorite, fun and affordance. Also, they should fill out a demographic questionnaire.
So we measured following dependent variables with the questionnaires:

- Motion Sickness Score
- Immersion Score
- User Experience Score
- NASA TLX Score
- Average individual elements score
- User Rankings

5.7 Procedure

After welcoming the participant to the study, he/she was informed about the rough procedure of the study. We explained that the study consists of two different

experiments and that the HTC Vive and the Leap Motion will be used for both of them. During the whole study, the communication took place in the mother tongue of the participant. An already prepared document with all the needed information for the first experiment was handed out and the participant should read it. By doing so, it enhances the consistency of the experiment and overall saved some time. After and during the read of the document, the participant was allowed to ask open questions. Then the participant was moved into the tracking area of the HTC Vive and got equipped with the HTC Vive HMD, which had the Leap Motion mounted in front of it [see Figure 5.1] and a pair of headphones. The volume of the used computer was set to the maximum value, and the application of the first experiment got started.
The participant spawned on a nearly empty virtual plateau in the sky. The only available things are a wall and the red interaction sphere, which was described in the study design. The user should position himself so that the red sphere was between him and the virtual wall. By doing so, it enhanced the visibility of all interfaces by having a neutral gray background [See Figure 5.2]

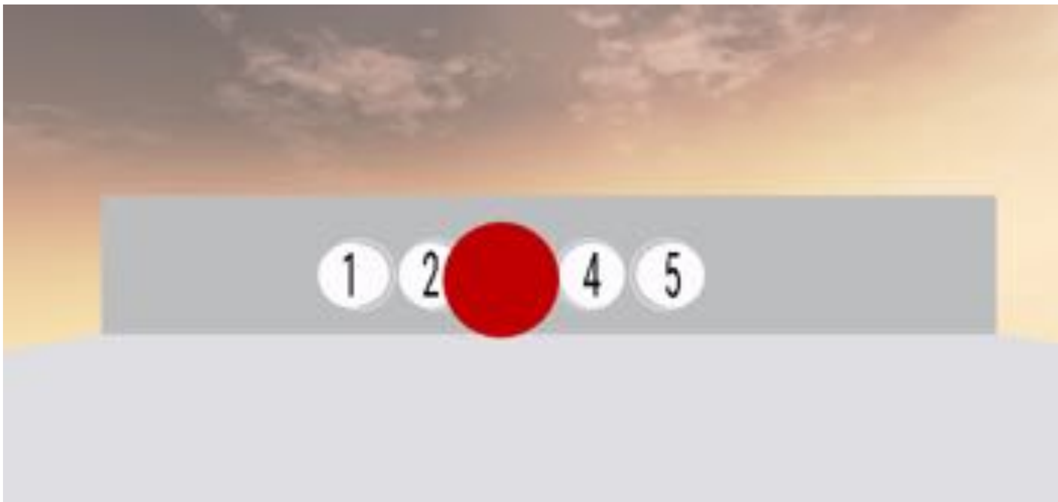


Figure 5.2: The gray wall, the sphere and an UI of Experiment 1

A training phase with the red sphere starts. The experimental subject got some time, to get familiar with it. It is important that the user knows how the sphere works because it is the trial activator for this experiment. On the participant's confirmation of having an understanding of the sphere, the conductor let the first interface appear and an exploration phase started.
Now the participant got some time, to get to know the interface. As soon as he felt

confident with it, the conductor spawned the red sphere and the experimental phase for the picked interface began. After the participant let the sphere disappear one of five buttons, which were labeled from 1 to 5, got marked with a blue color and the participant needed to interact with it. On interaction with any element, the elements became yellow/orange for a short time and played some feedback sound. If the participant interacted with the right element, the blue mark got removed and the sphere appeared again.
This cycle was repeated until the interface changed. Then another exploration phase with the new interface started, followed by the same experiment phase again.
After repeating the cycle for every interface, the first experiment got concluded with a questionnaire.
It is important to mention, that for both, the first and second experiment, the participant was standing. For all questionnaires, he/she was sitting.
The second experiment also started with an already prepared document, with all the needed information regarding the experiment. After finishing reading and confirming that the participant understood the most important parts of the document, he/she was moved into the tracking area again. The provided equipment was the same as in the first experiment: HTC Vive, Leap Motion (Front Mount on HTC Vive) and a pair of headphones.
For all of the 3 Interfaces iterations, they start with a training phase, where they learn to use the interface. The training was partially lead by the conductor, to ensure that the participant understood the most important aspects and differences between the interfaces.
After feeling confident, the participant should fulfill the 26 different tasks per interface iteration [see Table 5.3]. The conductor reads a task out loud and the participant confirms, that he/she understood the instructions. On confirmation, the conductor marks the given task as active and the participant can start. As soon as the participant thinks, that he/she solved a task, they should notify the conductor about it and the task will be marked as finished. This will be repeated for all 26 tasks.
To conclude one iteration, the participant should fill out a questionnaire and a NASA TLX and the iteration ends. This cycle will be repeated for all three interfaces.
When this is done the participant fills out two more questionnaires and the experiment is finally finished.

5.8 Results

In this results section and the following discussion we will make use of following abbreviations: 2DUI (Planar 2D User Interface), 3DUI (Spatial 3D User Interface), FCUI (Finger Count User Interface) and SD(Standard Deviation). We analyzed all experiments by using IBM SPSS Statistics 24. All significances were determined using either a univariate ANOVA or a multivariate ANOVA.

Experiment 1 Performance

Completion Time and Selection Accuracy:
Completion time was measured from the start of the task till the requested interaction with the interface. On average the 2DUI was the fastest interface with 0.67 seconds (SD=0.2) per interaction. The 3DUI was the second fastest interface with 0.82 seconds (SD=0.37) per interaction and the FCUI was the slowest interface with 1.61 (SD=0.89) seconds per interaction. We could find a significant difference in completion time between the three User Interfaces ($p < 0.001, F_{(2,981)} = 261.79, \eta^2 = 0.35$). We could not observe a significant difference in completion time between the three layouts (horizontal, vertical, circular) while comparing all three interfaces ($p < 0.5, F_{(2,981)} = 0.8, \eta^2 = 0.002$), but we could observe a significance when examining every interface individually, as you can see in Table 5.4.

Table 5.4: Completion Time

	2DUI		3DUI		FCUI	
	Mean	SD	Mean	SD	Mean	SD
Horizontal	0.68	0.17	0.9	0.42	1.54	0.64
Vertical	0.72	0.22	0.88	0.35	1.58	0.9
Circular	0.62	0.19	0.69	0.29	1.7	1.07
$p <$	0.001		0.001		0.4	
$F_{2,327}$	7.184		11.112		0.951	
eta^2	0.042		0.064		0.006	

For Values:	Best Value	Second Best Value	Worst Value
For Significance:	Significant	Significant Effect	No Significance

If we only compare the 2DUI and 3DUI we could still determine a significance regarding average completion time ($p < 0.001, F_{(1,654)} = 44.518, \eta^2 = 0.06$) but not regarding selection accuracy ($p < 0.8, F_{(1,658)} = 0.094, \eta^2 = 0$) as well.
Selection accuracy was determined by the unnecessary interaction between the start and end of the task. The average accuracy for the 2DUI was 95% (SD=28%), for the 3DUI it was 96% (SD=23%) and for the FCUI 86% (SD=41%). We could see a significance for selection accuracy between the three interfaces ($p < 0.001, F_{(2,981)} = 10.588, \eta^2 = 0.02$) but not between the different layouts ($p < 0.27, F_{(2,981)} = 1.309, \eta^2 = 0.003$) while comparing all three interfaces.
When looking at the layouts' selection accuracy of each interface individually, we could see no significance for the 3DUI and the FCUI. Only the 2DUI showed a significant effect. Especially with the circular layout, not a single unrequested interaction was made [see Table 5.5].

Table 5.5: Selection Accuracy

	2D		UI 3DUI		FCUI	
	Mean	SD	Mean	SD	Mean	SD
Horizontal	90%	43%	95%	23%	86%	35%
Vertical	95%	21%	94%	31%	87%	39%
Circular	100%	0%	99%	10%	84%	48%
$p <$	0.05		0.17		0.8	
$F_{(2,327)}$	3.66		1.78		0.238	
η^2	0.02		0.01		0.001	

For Values:	Best Value	Second Best Value	Worst Value
For Significance:	Significant	Significant Effect	No Significance

Completion Time and Selection Accuracy for the FCUI:
While examining the completion time for the FCUI we noticed a significance regarding requested numbers ($p < 0.001, F_{(4,325)} = 6.13, \eta^2 = 0.07$). To show and interact with the element one, the participants needed on average 1.35 seconds (SD=0.6), to show and thereby interact with element two they needed 1.77 seconds (SD=1.13), for element three 1.9 seconds (SD= 0.88), for element four 1.71 seconds (SD=1.11) and

for element five 1.31 seconds (SD=0.3) [See Table 5.6].
There is also significant effect regarding selection accuracy between the requested numbers ($p < 0.05, F_{(4,325)} = 3.33, \eta^2 = 0.04$). While interacting with element one the selection accuracy mean was 89% (SD=36%). For element two the accuracy rate was 79% (SD=48%), for element three 76% (SD=47%), for element four 87% (SD=46%) and for element five 99% (SD=12%) [See Table 5.6].

Table 5.6: Completion Time and Selection Accuracy of the FCUI per Finger

	Completion Time		Selection Accuracy	
	Mean	SD	Mean	SD
Showing 1 extended finger	1.35	0.6	89%	36%
Showing 2 extended finger	1.77	1.13	79%	48%
Showing 3 extended finger	1.9	0.88	76%	47%
Showing 4 extended finger	1.71	1.11	87%	46%
Showing 5 extended finger	1.31	0.3	99%	12%
$p <$	0.001		0.05	
$F_{(4,325)}$	6.13		3.33	
η^2	0.07		0.04	

For Values:	■	Best Values	■	3rd and 4th Best Value	■	Worst Value
For Significance:	■	Significant	■	Significant Effect	■	No Significance

Completion Time and Selection Accuracy for the 2DUI and 3DUI with regards to Extended Fingers:

As mentioned before, we are also observing how many fingers were extended during a successful interaction. Because the amount of extended fingers is given by the FCUI itself, we do only look at the 2DUI and 3DUI. For the 2DUI we could find a significance regarding completion time between different extended finger counts and for 3DUI we could at least see a significant effect [See Table 5.7].

Table 5.7: Completion time in Seconds

	2DUI		3DUI	
	Mean	SD	Mean	SD
0 extended fingers	0.63	0.02	0.83	0.03
1 extended fingers	0.7	0.02	0.79	0.03
2 extended fingers	0.74	0.04	0.83	0.11
3 extended fingers	0.59	0.05	0.72	0.11
4 extended fingers	0.76	0.05	0.89	0.1
5 extended fingers	0.67	0.04	1.27	0.14
$p <$	0.01		0.05	
$F_{(5,324)}$	3.2		2.72	
η^2	0.047		0.04	

For Values:	Best Values	3rd and 4th Best Value	Worst Value
For Significance:	Significant	Significant Effect	No Significance

The same also applies for selection accuracy. The 2DUI showed a significance regarding selection accuracy between different extended finger counts, but this time the 3DUI showed significance as well [See Table 5.8]

Table 5.8: Selection Accuracy

	2DUI		3DUI	
	Mean	SD	Mean	SD
0 extended fingers	96%	3%	93%	2%
1 extended fingers	99%	2%	100%	2%
2 extended fingers	81%	5%	90%	7%
3 extended fingers	100%	7%	100%	7%
4 extended fingers	93%	7%	92%	6%
5 extended fingers	83%	6%	71%	9%
$p <$	0.01		0.01	
$F_{(5,324)}$	3.1		3.25	
η^2	0.045		0.048	

For Values:	■	Best Values	■	3rd and 4th Best Value	■	Worst Value
For Significance:	■	Significant	■	Significant Effect	■	No Significance

Usability:
To measure usability and create huge comparability with Kulshreshth & LaViola [35]'s work, we asked the participants to rate their experience on a scale from 1 to 7 regarding overall appeal, mental demand, fatigue, pace of technique, selection rate, effort, frustration and difficulty. We found significance for every factor besides fatigue and effort when comparing the three interfaces [see Table 5.9]. When we only make a comparison between the 2DUI and 3DUI, we did not find any significance regarding any of the eight factors between them.

Table 5.9: Post Questionnaire Ratings

	2DUI Mean	2DUI SD	3DUI Mean	3DUI SD	FCUI Mean	FCUI SD	$p <$	$F_{2,30}$	η^2
Overall Best	5.27	1.56	5.72	1.42	3.36	1.69	0.01	7.12	0.322
Mental Demand	1.91	0.94	2	1	3.82	1.72	0.01	7.88	0.345
Fatigue	4	1.34	3.82	1.6	4.73	1.62	0.4	1.09	0.068
Pace of Technique	3.45	1.86	2.64	1.75	5.18	1.47	0.01	18.58	0.299
Selection Rate	5.91	0.83	6	1	4.45	1.37	0.01	6.96	0.317
Effort	6.73	0.47	6.73	0.47	6.73	0.47	1	0	0
Frustration	2.18	1.54	1.73	1.01	3.91	1.92	0.01	6.18	0.292
Difficulty	1.64	0.81	1.55	0.93	3.64	1.21	0.001	15.46	0.508

For Values:	■	Best Value	■	Second Best Value	■	Worst Value
For Significance:	■	Significant	■	Significant Effect	■	No Significance

Experiment 2 Usability

Completion Time and Error Rate:
For completion time we are only looking at the successfully solved tasks. All tasks which took more than 20 seconds to fulfill were marked as failed tasks and are not part of the average calculation. On average the tasks solved with the 2DUI took the participants 3.22 seconds (SD=0.24), with the 3DUI they needed 3.63 seconds (SD= 0.22) and with the FCUI the average time was 4.71 seconds (SD=0.25). But we could not find any significance for completion time. All three interfaces had exactly the same error rate of 7% (SD=0.015) and therefore we also saw no significance here as well.
Usability:
For usability we chose to use the User Experience Questionnaire (UEQ) by Laugwitz et al. [37] to measure user experience and obtain comparability with other systems, but we had to shorten it to reduce the length of the experiment. So we only concern one question per UEQ factor: Attractiveness, Efficiency, Perspicuity, Dependability, Stimulation and Novelty. The results are based on a scale between -3 and 3. The overall 2DUI user experience was rated at 1.136 (SD=0.82) on average. The 3DUI scored an average rating of 1.36 (SD=0.9) and the FCUI scored -0.11 (SD=0.95). There is a significant difference in user experience between the User Interfaces ($p < 0.01, F_{(2,30)} = 239.77, \eta^2 = 0.36$). When we look at the individual factors of the UEQ we can also see a significant difference between the User Interfaces for every

factor [Table 5.10].

Table 5.10: UEQ Factors

	2DUI Mean	2DUI SD	3DUI Mean	3DUI SD	FCUI Mean	FCUI SD	$p <$	$F_{(2,30)}$	η^2
Attractiveness	1.27	1.35	1.45	1.29	-0.45	1.21	0.01	7.38	0.33
Efficiency	1.09	1.51	1.18	1.35	-1.27	1.35	0.001	10.88	0.42
Perspicuity	2.27	0.79	2.09	0.54	-0.09	1.48	0.001	17.84	0.54
Dependability	1.09	1.38	1	1.34	-0.27	1	0.05	4.07	0.21
Stimulation	1.09	1.22	1.36	1.12	-0.27	1.35	0.01	5.56	0.27
Novelty	0	1.79	1.09	1.51	1.73	0.47	0.05	4.41	0.23

For Values:	■	Best Value	■	Second Best Value	■	Worst Value
For Significance:	■	Significant	■	Significant Effect	■	No Significance

To measure motion sickness, we prepared a shortened Motion Sickness Assessment Questionnaire (MSAQ) by Gianaros et al. [26], which concerned all four factors of the MSAQ (Gastrointestinal, Central, Peripheral, Sopite-related). We could not observe any significance regarding motion sickness between the interfaces ($p < 0.8, F_{(2,30)} = 7.638, \eta^2 = 0,018$). The FCUI achieved the highest motion sickness score (M=22.47, SD=14.88), followed by the 2DUI (M=19.7 SD=10.86) and the 3DUI with the lowest score (M=18.69 SD=11.59). Also, none of the four individual factors showed any significance at all [Table 5.11].

Table 5.11: MSAQ Factors

	2DUI Mean	2DUI SD	3DUI Mean	3DUI SD	FCUI Mean	FCUI SD	$p <$	$F_{(2,30)}$	η^2
Gastrointestinal	15.15	10.27	16.16	11.51	20.2	20.38	0.7	0.36	0.02
Central	25.25	21.14	22.22	19.88	28.28	24.53	0.8	0.21	0.14
Peripheral	15.15	10.27	17.17	11.51	16.16	10.23	0.9	0.1	0.06
Sopite-related	23.23	16.07	19.19	13.23	25.25	17.98	0.7	0.66	0.03

For Values:	■	Best Value	■	Second Best Value	■	Worst Value
For Significance:	■	Significant	■	Significant Effect	■	No Significance

Immersion and presence of the virtual world is not our main focus for system menus

because they will break the immersion by design. So we decided to just ask two question. One question concerned the overall immersion and participants should rate it on a scale from 1 to 7, with 1 being the worst and 7 being the best value regarding immersion. The other question was optional. The participants could describe which elements helped to create immersion and which elements destroyed immersion for them. All three UIs were rated nearly equally, with the 2DUI being rated with a value of 5 on average (SD=1.73), the 3DUI with a value of 5.09 (SD=1.76) and the FCUI with a value of 5.18 (SD=1.54). So we could not find significance regarding immersion between the different UIs ($p < 0.97, F_{(2,30)} = 0.03, \eta^2 = 0,002$).
To further investigate usability and gain better comparability we also used the NASA TLX [30]. The average total workload for the 2DUI was 43.73 (SD= 21.4). The 3DUI scored on average 35.67 (SD= 18.47) and the FCUI 59.58 (SD= 19.12). There is a significant effect regarding the NASA TLX score and the interfaces ($p < 0.05, F_{(2,30)} = 4,19, \eta^2 = 0.218$). Two of the individual factors of the NASA TLX showed significance, one had a significant effect and three factors showed no interaction at all [See Table 5.12].

Table 5.12: NASA TLX Scores

	2DUI Mean	2DUI SD	3DUI Mean	3DUI SD	FCUI Mean	FCUI SD	$p <$	$F_{(2,30)}$	η^2
Mental Demand	29.55	19.16	25.91	13.19	60.45	23.07	0.001	11.08	0.425
Physical Demand	45.91	27.09	42.27	20.29	57.27	23.38	0.3	1.19	0.074
Temporal Demand	40	26.46	31.36	15.83	54.09	25.48	0.08	2.72	0.153
Performance	41.82	23.16	38.18	24.11	56.82	24.32	0.17	1.88	0.112
Effort	46.82	23.59	42.27	19.92	64.55	17.53	0.05	3.63	0.195
Frustration	43.18	28.04	30.45	19.68	62.73	21.37	0.01	5.348	0.263

For Values:	Best Value	Second Best Value	Worst Value
For Significance:	Significant	Significant Effect	No Significance

To better analyze the interfaces' different representations of the individual elements [see chapter 3], we asked the participants to give an overall rating of the elements ranging from 1 to 7, with 1 being the worst and 7 being the best rating [See Table 5.13].

Table 5.13: Individual Element Rating

	2DUI Mean	2DUI SD	3DUI Mean	3DUI SD	FCUI Mean	FCUI SD	$p <$	$F_{(2,30)}$	η^2
Menu Navigation Buttons	5.27	1.56	6	1.27	3.45	1.51	0.001	9.017	0.375
Name Input Widget	5.36	0.81	5.09	1.58	3.09	1.3	0.001	10.508	0.412
Height Input Widget	5.45	0.69	5.91	1.22	3.09	1.51	0.001	17.756	0.542
Birthday Input Widget	5.73	0.79	5.09	1.64	1.82	1.25	0.001	29.795	0.665
Skill Distribution Widget	5.64	1.12	5.64	1.29	3.45	1.44	0.001	10.511	0.412
Light & Music Switch	5.64	1.03	5.45	1.7	3.91	1.14	0.01	5.697	0.275
Hue & Volume Slider	3.73	2.15	5	1.84	5.09	1.22	0.15	2.017	0.119
Track Switcher	5.73	0.79	5.73	1.42	4.45	1.57	0.05	3.488	0.189
Graphics Quality Switcher	5.73	1.01	5.73	1.19	3.73	1.67	0.001	8.705	0.367
Advanced Settings Switches	5.73	0.79	5.73	1.19	4	1.67	0.01	6.786	0.311
Menu Color Slider	4.18	1.94	4.91	1.92	4	1.48	0.5	0.791	0.05
Over Interface Rating	5	1.41	5.55	1.44	3.36	1.21	0.01	7.697	0.339

For Values:	■	Best Value	■ Second Best Value	■ Worst Value
For Significance:	■	Significant	■ Significant Effect	■ No Significance

It is important to mention that we could not find a single significance between the 2DUI and 3DUI regarding any of the observed dependent variables in our second experiment.
To investigate something as abstract as fun, we asked the 11 participants to assign a rank from 1 to 3 for each interface regarding fun, affordance and overall appeal. Rank 1 should be assigned to the best and rank 3 to the worst interface. The 3DUI scored best regarding fun (Rank 1: 8 Votes; Rank 2: 1 Vote, Rank 3: 2 Votes), followed by the 2DUI (Rank 1: 3 Votes; Rank 2: 6 Vote, Rank 3: 2 Votes) and the FCUI ranked last (Rank 1: 0 Votes; Rank 2: 4 Vote, Rank 3: 7 Votes) [see Figure 5.3].

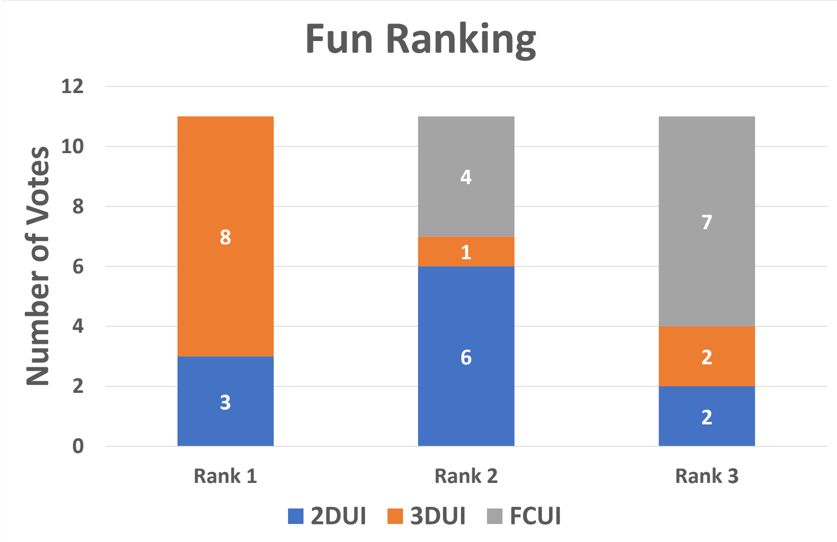


Figure 5.3: Questionnaire Results regarding Fun

It is hard to determine a clear winner regarding affordance. The 2DUI (Rank 1: 5 Votes; Rank 2: 6 Vote, Rank 3: 0 Votes) is slightly ahead the 3DUI (Rank 1: 6 Votes; Rank 2: 3 Vote, Rank 3: 2 Votes) when calculating the mean, but still the 3DUI was ranked more often on rank 1. The clear loser is the FCUI (Rank 1: 0 Votes; Rank 2: 2 Vote, Rank 3: 9 Votes) [see Figure 5.4].

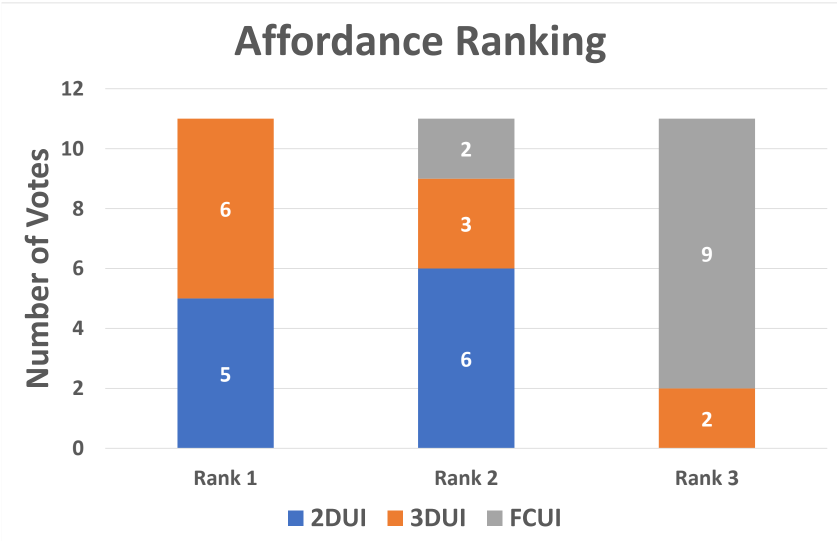


Figure 5.4: Questionnaire Results regarding Affordance

Overall appeal was more clear. The participants liked the 3DUI the most (Rank 1: 7 Votes; Rank 2: 3 Vote, Rank 3: 1 Votes), the 2DUI the second most (Rank 1: 4 Votes; Rank 2: 5 Vote, Rank 3: 2 Votes) and the FCUI the least (Rank 1: 0 Votes; Rank 2: 3 Vote, Rank 3: 8 Votes) [see Figure 5.5].

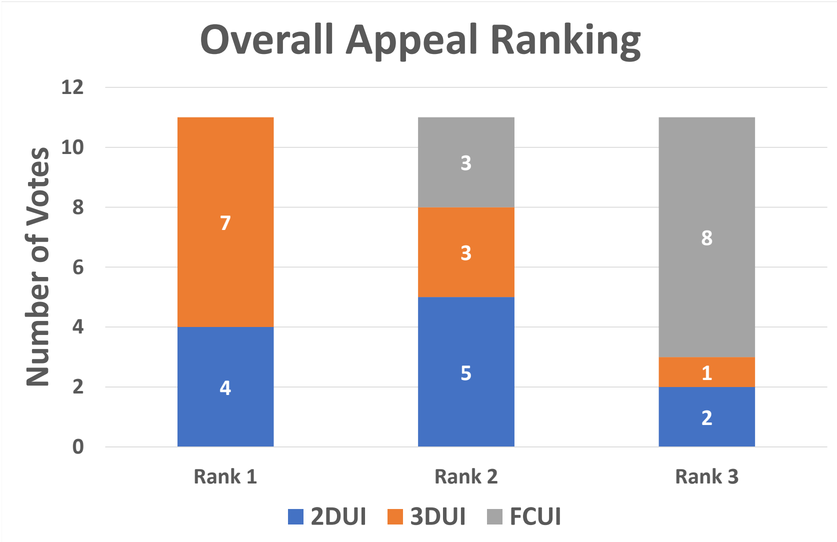
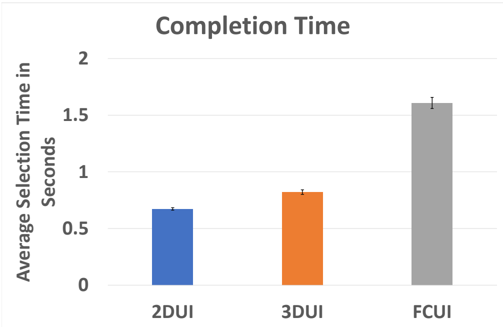


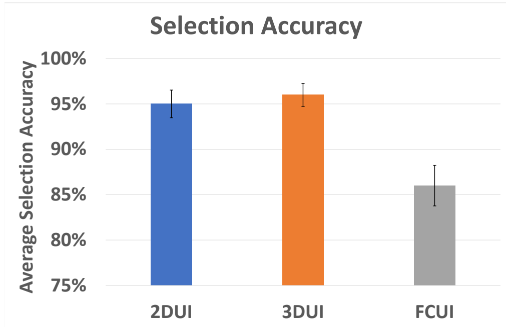
Figure 5.5: Questionnaire Results regarding Overall Appeal

5.9 Discussion

5.9.1 Experiment 1 Performance



(a) Completion Time



(b) Accuracy

Figure 5.6: Completion Time and Accuracy of the different UIs

Performance wise the FCUI cannot compete with the 2DUI and 3DUI. It is significantly slower and less accurate [see Figure 5.6]
The bad time performance can be explained by the dwell time. To interact with a Finger Count Button one does need to extend a certain amount of fingers for 0.5 seconds. So the best time one could achieve with the FCUI would be 0.5 seconds which is still impossible, because you still need reaction time to identify the element which needs to be triggered and time to actually extend the equivalent amount of fingers. So we get following equation regarding Finger Count interaction time:

$$InteractionTime = 0.5 + ReactionTime + ResponseTime$$

To come closer to the average time of the 2DUI (0.67 seconds) the reaction time and response time needs to be smaller than 0.17 seconds, which cannot be achieved by humans [48]. The only way to get comparable times would be by decreasing the dwell time, which would lead to even less accuracy for the FCUI.
The bad accuracy can be explained by two factors:

1. The dwell time of 0.5 is too short-> when unintentionally interacting with an element 0.5 is sometimes too short to notice the upcoming interaction and then to also cancel it by changing the number of extended fingers in this small time frame.

2. An optical input device like the leap motion [Quelle] is not accurate enough. When fingers are too close together, they are sometimes only seen as one. Or when one finger covers another you will get the same effect. So some postures work better than others.

The 2DUI is the fastest interface (H_2). It is significantly faster than the 3DUI. A possible explanation could be that one needs fewer steps to interact with the buttons. When interacting with the 2DUI, one just needs to fulfill one step, which is touching the button. To interact with the 3DUI users need to reach out their hand in front of the hovering button and apply force so that it collides with the trigger plane [see chapter 3.4]. So we have one working step against two working steps. This could also be the reason why the 3DUI has slightly higher average accuracy than the 2DUI, although it is not a significant difference.

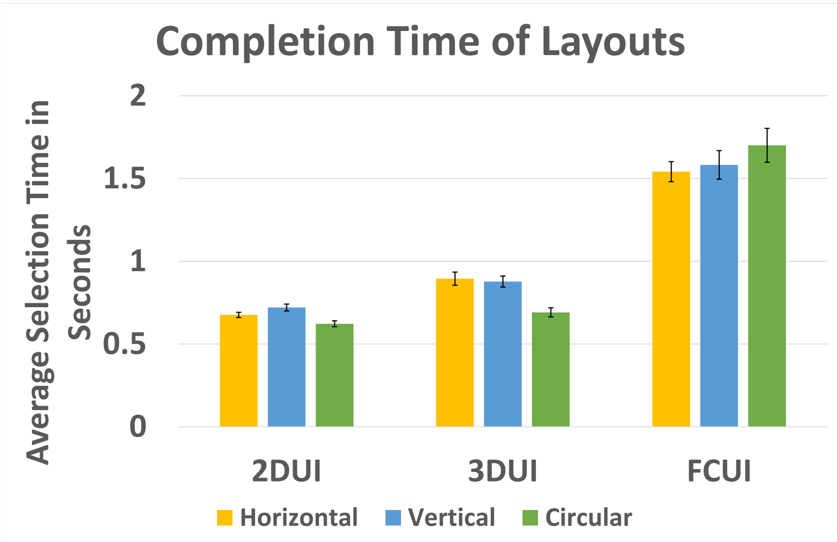


Figure 5.7: Time needed to interact with the different Layouts

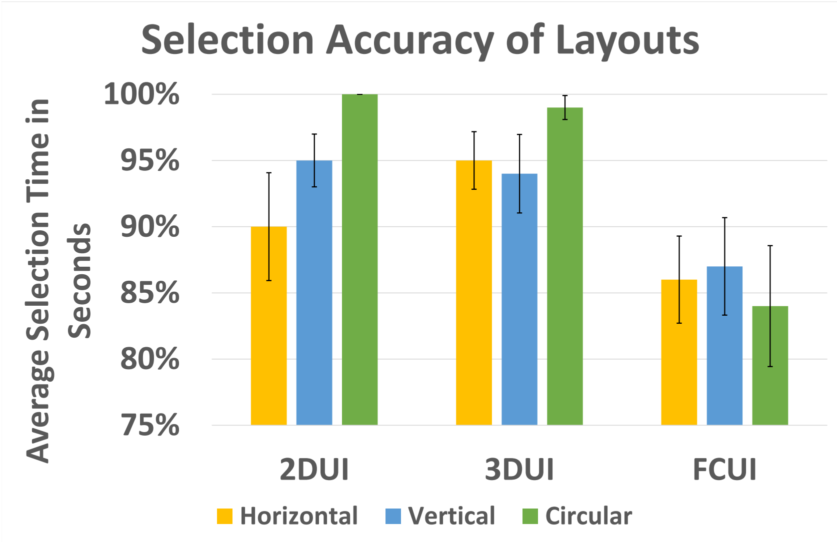


Figure 5.8: Accuracy while interacting with the different Layouts

The circular layout is the fastest and most accurate layout for the 2DUI and 3DUI [see Figure 5.7 and Figure 5.8]. The hand has the same distance to every button which reduces the average completion time and there is far more space in between the buttons which significantly increases the accuracy. Those results disconfirm H_3. The horizontal layout was the fastest and vertical was the most accurate layout for the FCUI. This results can be explained by cultural constraints. In Germany texts and enumerations will be read from left to right and from top to bottom. So it is easier for the participants to map the location of a button to a number in the horizontal and vertical layout. With this mapping, the participants subconsciously do not need to read the labels on the buttons every time. Due to the fact, that the difference in speed and accuracy between the different layout of the FCUI is not significant, this could also just be a random occurrence.
From all observed combinations, the 2DUI with the circular layout is the fastest and most accurate one, closely followed by the 3DUI.

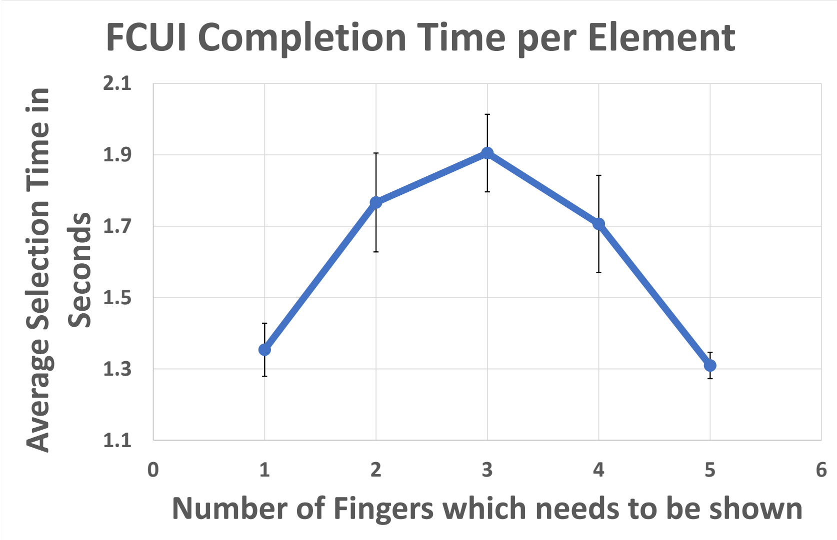


Figure 5.9: Time needed to interact with different FC Buttons

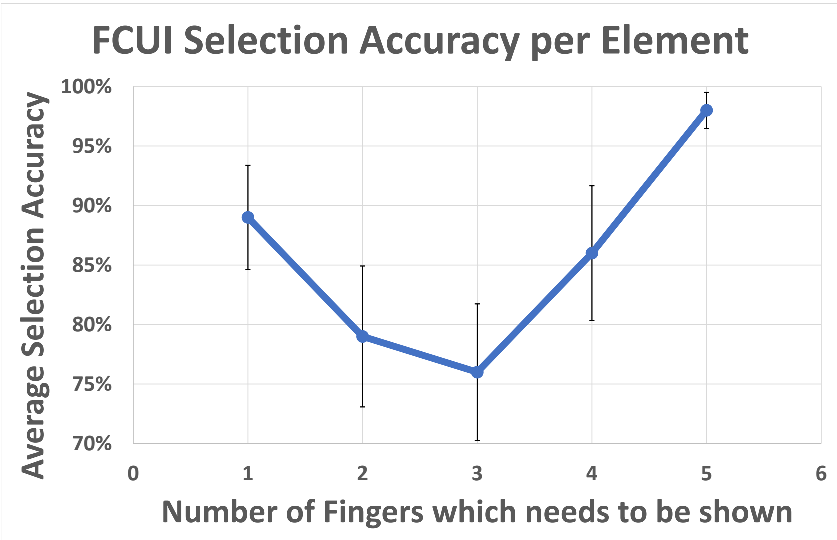


Figure 5.10: Accuracy while using different FC Buttons

While using the FCUI the elements with the numbers 1 and 5 are the fastest and most accurate, with the 5 achieving even better performance than the 1 [see Figure 5.9 and Figure 5.10]. When interacting with element 5, there is just one possible combination of extended fingers to show a five (when only using one hand, which was obligatory in the first experiment). So users do not need to make a decision which posture to take when there is only one. Also when showing a five, participants spread out the fingers further apart from each other, which helped the optical system to recognize the shown posture even better. Those two factors could explain the best speed and accuracy among the Finger Count elements. Element 1 performed slightly worse regarding speed because there are five possibilities to show a one. The worse accuracy emerges because some participants decided to extend their index finger to show a one, but they also slightly extended their middle finger during this process which the camera recognized as two extended fingers. It is important to mention that it is just natural to do such a slight extension of the middle finger and it is physically demanding to avoid it.

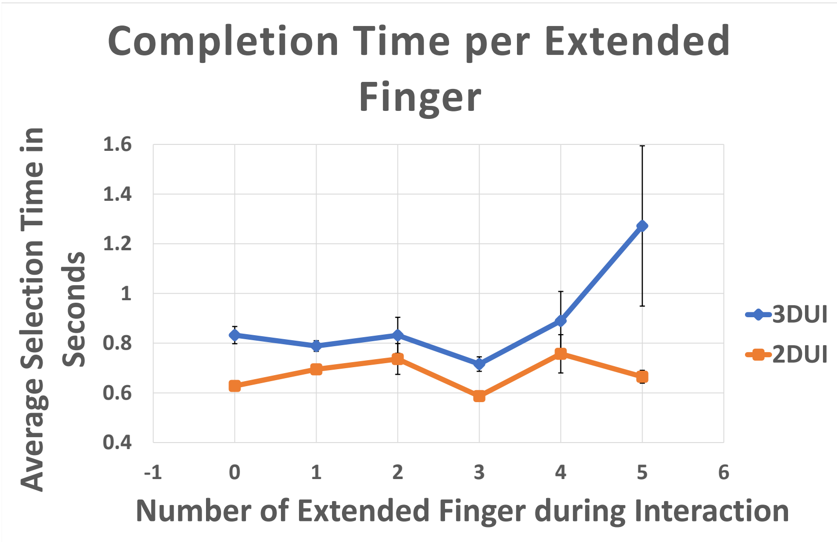


Figure 5.11: Time needed for 2D Button or 3D Button interaction per Hand Posture

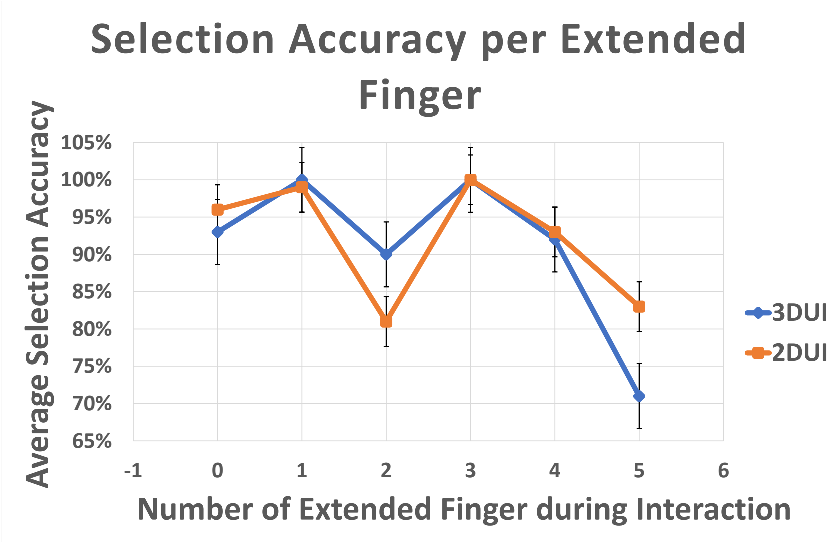


Figure 5.12: Accuracy during 2D Button or 3D Button interaction per Hand Posture

While interacting with the 2DUI, it seems to be the fastest way to use three fingers [see Figure 5.11]. For example, interactions, where the participant's thumb, index and middle finger, or their index, middle and ring finger were extended, were the fastest on average, closely followed by the ones where a fist posture was used. Interactions where three fingers where extended were also the fastest for the 3DUI, but the second fastest where the interactions triggered while having only one finger extended (mainly the index finger).

Having three fingers extended during the time of interaction, was the most accurate way of interaction for the 2DUI and the second most precise way for the 3DUI [see Figure 5.12]. Having one finger extended was the second most accurate way of interaction for the 2DUI and the most accurate one for the 3DUI.

It is hard to explain why extending three fingers during the interaction caused the best results. The only causality we came up with, is that the size of the three extended fingers hitbox could be the sweet spot between being too big to accidentally triggering other buttons and being too small to consume to much time to aim at the buttons or even missing them.

Also, take the results regarding the number of fingers extended during the time of interaction with a grain of salt. The collected samples are not equally distributed. We collected far more interactions with zero to two fingers than interactions with three to five extended fingers. One would need to redo the experiment and control the number of extended fingers so that all hand postures were used equally often.

OB = Overall Best	MD = Mental Demand	FAT = Fatigue	PT = Pace of Technique
SR = Selection Rate	EF = Effort	FRU = Frustration	DIF = Difficulty

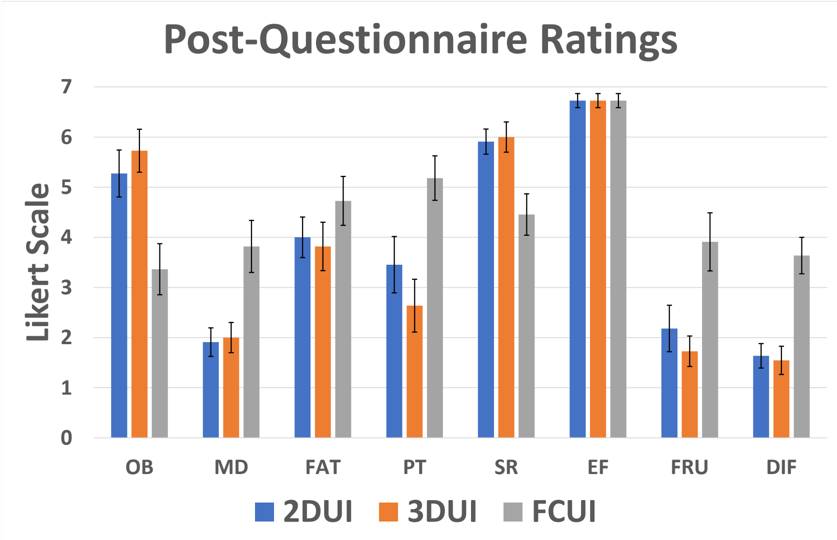


Figure 5.13: Post-Questionnaire Rating after the first Experiment

The FCUI obviously achieved the worst results in the post-questionnaire [see Figure 5.13]. Participants gave informal feedback, that it is hard for them to map the numbers to a certain hand posture while not being able to see their real hand, which could be a reason for the high mental demand. Some also mentioned, that it is tiring to switch between hand postures in such a high frequency, which causes a higher rating of fatigue. The short dwell time of 0.5 seconds causes a feeling of constant pressure which could lead to the bad rating of the technique's pace. The participants noticed that the Leap Motion [4] misinterpreted their hand posture sometimes, which lead to interactions with unrequested elements, which finally lead to a bad rating in selection rate, frustration and difficulty. The higher difficulty rating was also related to the higher mental and physical demand. During the use of all three UIs, the participants put in the same effort.
The 3DUI was the best-rated interface (H_1). It was rated better than the 2DUI in every category besides mental demand. As already mentioned before, fulfilling two working steps mentally more demanding than just fulfilling a single working step. The better fatigue rating could be explained by the fact, that the movable

disc part of the 3D Buttons was closer to the user than the 2D Button itself and it could be launched into the button's trigger plane [see chapter 3.4] without needing to move the hand the whole way. It is hard to find a good reason why the 3DUI has a better pace of technique than the 2DUI. Maybe one could compare it to an actual object which could move but does not, e.g. a vase with flowers or a stone. Looking at such objects which are standing still could also calm down the persons looking at it. The better frustration rating of the 3DUI compared to the 2DUI can be explained by the additional working step of the 3DUI. When touching a 2D Button, users cannot cancel the interaction and will definitely cause unintended interaction when touching a wrong button. As long as the movable disc part does not touch the trigger plane, users can cancel the interaction. So the additional working step functions as a protective mechanism but sacrifices some speed for it.

5.9.2 Experiment 2 Usability

Testing the different interfaces in a realistic scenario and with all necessary widgets showed that there is no significant difference between speed and error rate among them. Still, the 2DUI was the fastest followed by the 3DUI and then by the FCUI.

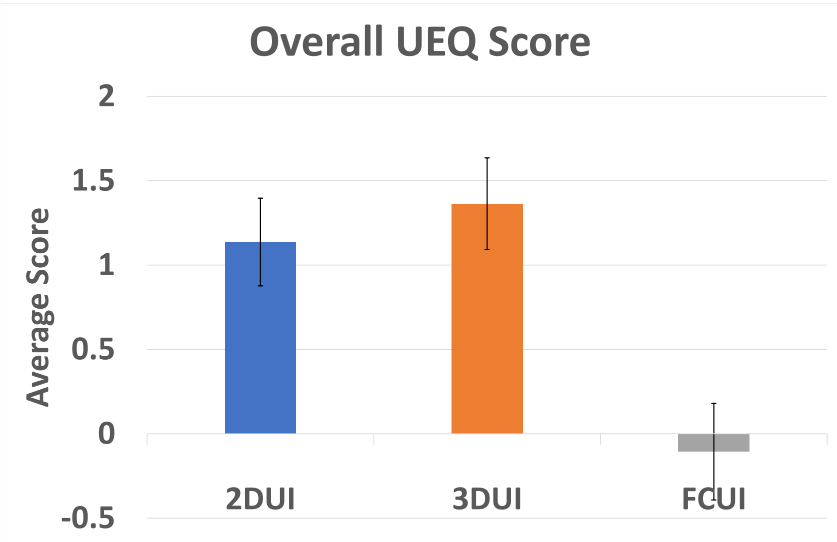


Figure 5.14: The overall UEQ Score of the three different UIs

ATT = Attractiveness	EFF =Efficiency	PER = Perspicuity
DEP = Dependability	STI = Stimulation	NOV = Novelty

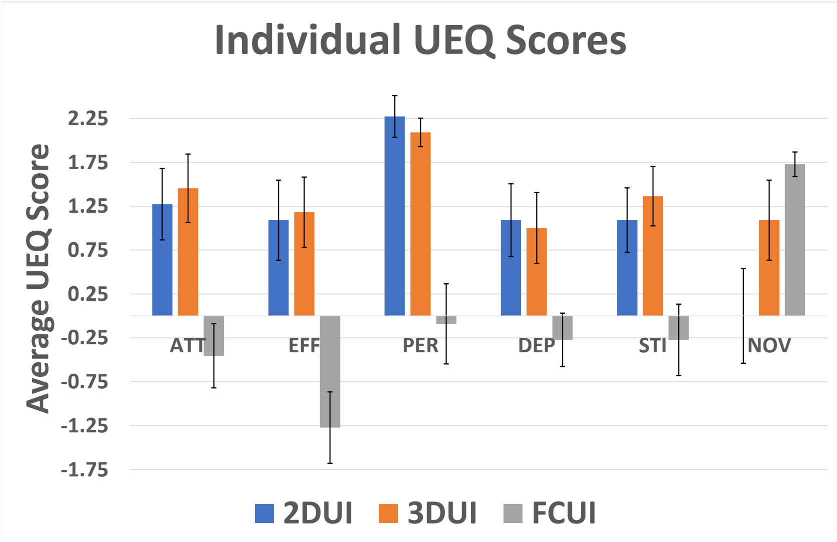
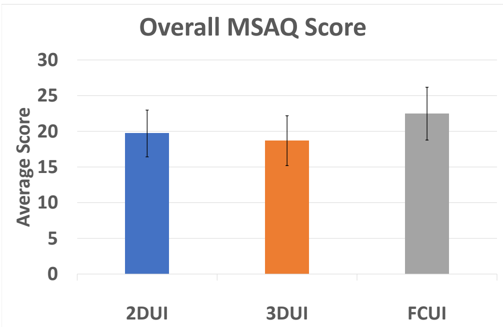


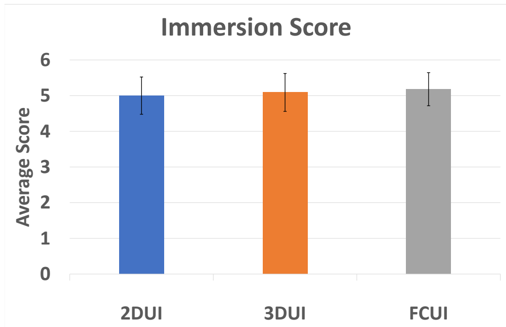
Figure 5.15: The UI Ratings of the UEQ's individual Factors

When looking at the UEQ scores, we can clearly see that the FCUI provided the least usability [see Figure 5.14]. When interpreting the FCUI's individual UEQ scores using the UEQ Benchmark [43], every UEQ factor would be classified as bad [see Figure 5.15]. Novelty is the only exception. The novelty score equals an excellent rating on the benchmark, which is not very surprising because the idea of the Finger Count Menu is rather new and not used outside of research facilities yet.
The 2DUI and the 3DUI are pretty similar and it seems like a neck and neck. Still, the 3DUI has the best overall UEQ score (H_1). With the help of the UEQ Benchmark [43], we can better differentiate between the 3DUI and the 2DUI regarding the single UEQ factors [see Figure 5.15]. The 3DUI as well as the 2DUI, have an above average attractiveness, an above average efficiency and an excellent perspicuity. Regarding perspicuity, it is worth mentioning, that the 2DUI archived a better score than the 3DUI, which gives grounds for assuming, that the familiarity gained from everyday devices like smartphones and the declining use of analog 3-dimensional devices in today's society plays an important role here.

The 2DUI has an above average and 3DUI a below average dependability according to the UEQ Benchmark. This could also be observed in the experiment. Some participants had problems in understanding that all the 3DUI elements were only pushable and not draggable. Also sometimes they moved their hand too fast towards a 3DUI element so that the system could not apply force on the element fast enough and the hand slipped through it.
The 3DUI has an above average and the 2DUI a below average stimulation rating. This means participants found it more exciting and motivating to use the 3DUI then the 2DUI. The affordance oriented design of the 3DUI could be one reason for this outcome. Another reason could be that the 3DUI is something more innovative and creative, as we can also see in the novelty rating. The 3DUI's novelty rating is classified as good by the benchmark and the 2DUI's rating as bad. It seems like 2D interfaces are such a huge part of today's society, that people forgot about the old 3-dimensional analog interfaces and think it is innovative to use them again.



(a) Motion Sickness



(b) Immersion

Figure 5.16: MSAQ and Immersion Rating of the different UIs

As we can see, there is nearly no difference in motion sickness and immersion between the three interfaces [see Figure 5.16]. The motion sickness score and immersion scores are both in a range where users should not feel any discomfort and have a good virtual presence. We provided a small furnished apartment as a virtual environment, which we think is the more important factor for immersion here than the interface. A system menu breaks the immersion by design because one wants to alter values which are hardly implemented into the virtual world, e.g. altering the overall volume or leaving the current application.

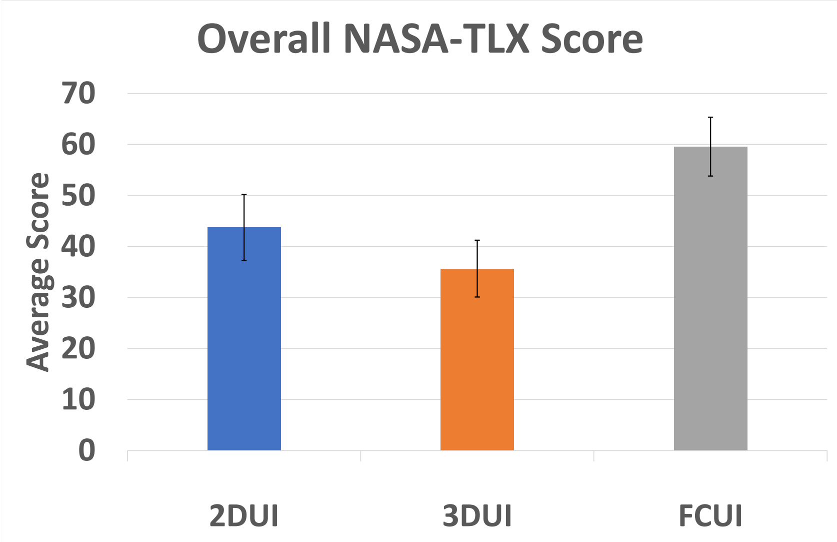


Figure 5.17: Overall NASA-TLX Score of the different UIs

MD = Mental Demand	PD =Physical Demand	TD = Temporal Demand
PER = Performance	EFF = Effort	FRU = Frustration

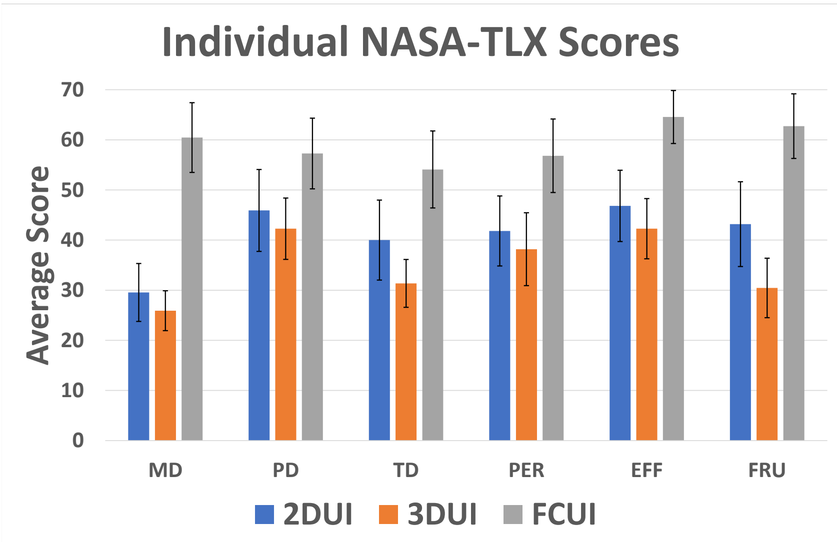


Figure 5.18: The individual NASA-TLX Factor Scores of the UIs

When looking at the total workload of the NASA TLX, we can see that the 3DUI was rated best (H_1) and the FCUI was rated worst [see Figure 5.17]. The 3DUI was also rated best in every single factor of the NASA TLX (H_1) and the FCUI was rated worst [see Figure 5.18]. As we already stated for the first experiment, when using the FCUI it is hard to map every number to a certain posture and also from the posture while not being able to see the real hand. This could be the reason for the high mental demand of the FCUI. The explanation for the results of all other factor is the same as in the first experiment [see chapter 5.8]. For frustration, it is worth mentioning, that many participants had huge problems using the 2D Slider. Entering a precise value with it is hard. When releasing the 2D slider, micromovement caused a change of the entered value in the last moment. This was perceived as very frustrating by the participants and affected the overall experience with the 2DUI negatively. It is worth mentioning that the 3DUI had the same issue, but not as bad as the 2DUI.

MNB = Menu Navigation Buttons	NIW = Name Input Widget	HIW = Height Input Widget
BIW = Birthday Input Widget	SDW = Skill Distribution Widget	LMS = Light & Music Switches
HVS = Hue & Volume Slider	TS = Track Switcher	GQS = Graphic Quality Switcher
ASS = Advanced Settings Switches	MCS = Menu Color Slider	OIR = Overall Interface Rating

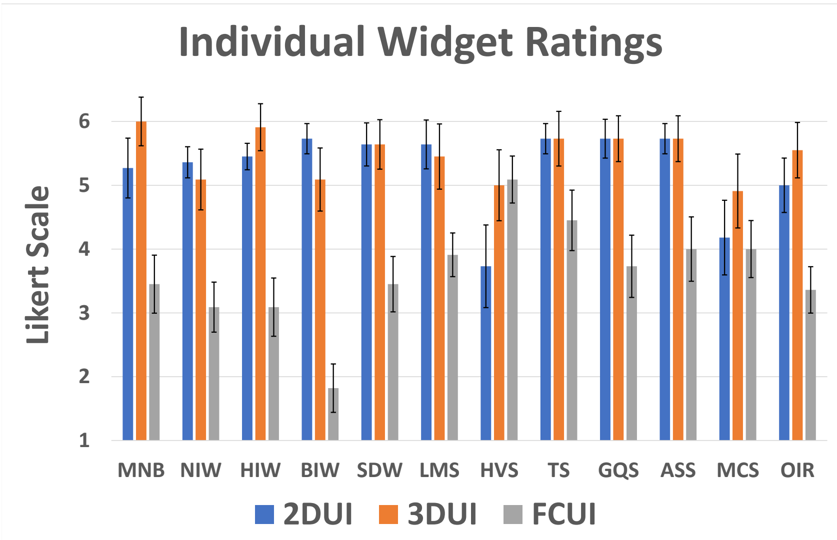


Figure 5.19: The Ratings of the Menu's Widgets in their respective Representation

Regarding the individual widget ratings, none of the results between 2DUI and 3DUI were significant. So take the following discussion about the different widgets with a grain of salt, because we only concern the average ratings here. Further experiments with more participants need to be conducted to clear this matter. At first sight, one can see from Figure 5.19 that the widgets of the FCUI were all rated significantly inferior to the 2DUI and 3DUI Widgets. Only the FC Fixed Slider could stand out as you can see in the HVS rating. With the MNB and the HIW, we can see the participants seem to like the 3D Buttons a bit more than the 2D Buttons. But the difference is rather small as one can see from the SDW and the GQS. The NIW and the BIW ratings are very surprising. They show that there is a slight preference for the 2D Buttons than for the 3D Wheel Picker. It seems like participants do not like to move their arms a lot and would rather want to let their hand rest at a certain position. The LMS show that participants like the 2D Buttons a bit better than the 3D Rocker Switches. Maybe because it is a rather trivial task and they do not want to take an extra step for it. The HVS shows that the 3D Slider is better than the 2D Slider. This is not surprising because in the experiment a lot of participants faced the same problem with the 2D Slider. When they tried to release the 2D Slider's marker, micromovement caused a change of the current value. This is very relevant in this experiment because the task demands precise input. It would have been interesting also to include a task that requires imprecise slider values. For example: "Set a value between 10 and 20". The FC Fixed Slider was the most preferred representation here because it was sufficient to show a single number to solve the task. The TS shows that participants like the 2D Buttons and the 3D Slider equally. But it is important to mention, that the standard deviation of the 3DUI is larger than the standard deviation of the 2DUI. The ASS show that the 2D Buttons are on par with the 3D Rocker Switches. The MCS confirms that the 3D Slider is more popular among the participants than the 2D Slider. In contrast to the FC Fixed Slider, the FC Dynamic Slider was rated pretty poorly. The most interesting result can be derived from the OIR. Although the difference between the 2DUI and 3DUI was rather small, the participants like the 3DUI the most. On average the 3DUI scored 0.5 points more than the 2DUI on the 1-7 Likert Scale.

From the final ranking task of experiment 2 we can see that the 3DUI is the interface which provides most fun to the participants (H_1). The reason for this could be the hedonomic design of the 3DUI. With it is simple and neutral coloring, it creates aesthetic longevity and because of the natural transitions the 3DUI provides by using movable elements, it also enables seamless interactions as Hancock et al. [29] recommends.

Affordance-wise it is not possible to declare a clear winner from the ranking but

we can see that the FCUI was the least liked interface. We noticed sometimes participants still tried to interact with the FCUI by touching it. Also sometimes the participants had problems to identify which number to show to interact with a certain element. Those two factors could be the reason for the bad ranking.
While looking at all different aspects of the interfaces and giving an overall subjective rating, most of our participants would prefer to use the 3DUI and would not want to use the FCUI (H_1).
In the end, we want to address that the fun, affordance and overall ranking can only compare the three tested interfaces and one can not make any conclusions regarding other interfaces. The main purpose of the ranking was to measure abstract concepts as fun and affordance. Also, we wanted to see whether the user's perception matches our statistical data, which we could confirm.

6 Design Guidelines for Mid-air Menu Control

Use the 2DUI with a circular layout for best performance

For creating the fastest and most accurate UI one should use the 2DUI with a circular layout as a starting point. It achieved the fastest completion time and the highest accuracy during our first performance experiment. If sacrificing a very small portion of speed and accuracy is possible, then one should rather use the 3DUI instead. With a circular layout it delivers slightly less performance but a lot more usability as our first and second experiment show.

Use the 3DUI for best usability and user experience

The 3DUI provided the best UEQ score, the best immersion score, the best MSAQ score, the best NASA TLX score, the best fun ranking and the best overall ranking in our experiments. It got a lot of positive informal feedback from users and they seemed to be more engaged during the use of it.

Avoid using a lot of two, three and fours in the FCUI

Our results show, that showing a two, three or four during the use of the FCUI, caused the longest completion times and most accuracy issues. Showing a five was the most reliable and fastest number, followed by the one.

Sliders and other movable objects need fixation on release

As we saw during our experiment, removing the hand from the slider is a crucial moment, which causes a lot of frustration. Solving this problem will most likely enhance the overall user experience a lot and is worth the implementation time. As we already mentioned in the concept of the 3D selection wheels, we use a function which changes the drag of objects depending on their current velocity. When an object is nearly not moving at all, it has higher drag to avoid micromovement. From higher velocity we can conclude, that the interaction is intentional and we lower the drag to support the users intentions. The drag's maximum and minimum are capped.

Disable other elements near the location of interaction

When interacting with a certain element, it helps a lot to make other elements which are close to the current interaction uninteractable. This is especially important in a vertical layout. For example, when interacting with a button using an extended index finger, it was pretty common during development to accidentally press the button below with the rest of the hand. This lead to a lot of frustration and unintentional interactions. Therefore we decided to disable the UI elements which were pretty close to the element of interest. To decide with which element the user wants to interact, is pretty tough. We tried to predict it depending on the position of the finger tips, but there are far to many possible hand postures to cover every case. We could have forced the users to use a certain posture, but in the end this will only limit the users and affects the experience of users who would rather prefer a different hand posture. In the end we decided to use a very short timer which is explained in the implementation chapter 4. It is important to mention, that one should not make all other elements uninteractable when developing a UI which can be used with both hands. After interacting with a certain element, users could want to interact with another element at nearly the same time or shortly after.

Handle cases where menu spawns inside the hand

During early pilot test we noticed that users who used both hands, often use one hand to switch through the different submenus and the other hand to interact with the content of the submenu. When they switched the submenu they tended to leave the hand which they used for interaction, at the location where they last used it. When switching menus, it sometimes happened that the new submenu had elements which intersected with the interaction hand and thereby will lead to an unintentional interaction. This is an edge case which can be handled pretty easy, by disabling interaction when such an incidence occurs and reenabling it, when the hand stops intersecting the submenu. This does not only concerns submenus, but also main menus when they can opened and closed at will.

Do not forget to handle the fact that multiple interactions are possible at the exact same time

In contrast to wimp-interfaces it is possible to interact with multiple elements at the same time with both hands. This creates a huge space of new problems, especially when using Unity [13] you have to be careful. For example it is possible to close a submenu while interacting with an UI element of it. Developers need to handle the the last known state of the UI element and therefore create and use a function which is called before actually closing the submenu. When developing with Unity, developers should outsource the functionality of elements which can potentially be disabled, because disabled objects are not able to do anything anymore. For example, we put some audio sources on the close buttons of the different submenus, but we could not hear a sound when using them. Directly after pressing one of them, the submenu and all elements got disabled and thereby also their whole functionality and their audio sources.

Audio feedback is essential during the absence of haptic feedback

With Chan et al. [23]'s work and our own experience we have grounds to suspect, that the lack of sound and any other visual aids will make it harder for the users to estimate distances to virtual objects and thereby decrease the overall performance. Take this with a grain of salt, because we need to verify whether those findings still apply for stereoscopic HMDs in future work.

Be aware of the Midas Touch Problem for the FCUI

When creating any Finger Count application, it is important to be aware of the fact, that it is possible to show numbers from zero to ten. So we have eleven possible inputs, from which one needs to be assigned to no interaction. Otherwise, users cannot rest and every input would trigger an event.
Some similar phenomenon happens, when the Finger Count application supports menu navigation. Let us assume, one could enter a given submenu by showing a two and this submenu contains another submenu which can also be accessed by showing a two. Then again in this sub-submenu, there is an interface element which can be triggered by showing a two as well. In early development, by showing a two to

open the submenu, the sub-submenu was also opened and an interaction with the mentioned element occurred in the same instance. This was caused by only using a single timer, not resetting the timer after an interaction and by only using the timer value as a condition. We included the reset function of the timer value but the interaction still felt odd. Upon showing a two now, just the submenu would open but directly after opening it, the progress bar to open the sub-submenu would start to fill up without giving the user any chance to explore or look at the submenu first. This is fatal with very short dwell times. So we also included a check for this. In the final prototype, continuing to show a two directly after opening the submenu would do nothing and would give users enough time to explore the submenu first. If one still wants to open the sub-submenu by showing a two, it is obligatory to show a number which is different from the one which was used to open the submenu, first.

7 Conclusion

In this thesis, we researched mid-air interactions for menu control. Therefore, we implemented three different User Interfaces with three completely different key aspects: simplicity (2DUI), affordance (3DUI) and novelty (FCUI). Two interfaces are using direct manipulation as interaction mode, which is well known from touchscreens nowadays. The third interface uses its distinctive Finger Count interaction mode, which utilizes the ability to count with the fingers of the hands. All hand movements were captured and analyzed with a Leap Motion sensor which was attached to an HTC Vive. During our work we analyzed many related works, to get more insight on fundamental principles of 3D UI design, which techniques are available and what we can learn and adapt from other UIs. The Finger Count technique arose from those investigations. Originally it was used in a desktop environment and in this thesis, for the first time, it was implemented and tested in VR and the context of menu control. The affordance oriented 3DUI made use of the combination of isomorphic and nonisomorphic approaches to boost usability, whereas the simple and plain 2DUI refrains from any technique that extends the gulf of execution. All implemented UIs in this thesis were not meant to be a finished product. They should serve researchers and UX Designers as a baseline and show what can be expected from the introduced approaches. Also, our work should encourage them to think out of the box and to push the boundaries of what is possible in VR. VR applications are self-contained and one should put as much work into the menu control as in every other part of the application, in order to not break immersion or the overall user experience. To provide this baseline, two separate experiments were conducted to measure performance and usability of our UIs. In the first experiment, eleven participants had to use the three different UIs and three different layouts for menu selection. In the second experiment, a realistic scenario and environment was given to the participants and they should use three fully functional system menus, based on the three UI approaches, to fulfill typical menu tasks. Completion time, hand posture, error rate and accuracy were tracked during both experiments. Also, after both experiments, the participants filled out questionnaires about user experience, task load, motion sickness, immersion, individual UI widget rating and demographic data. The gathered results showed that the affordance-based 3DUI achieved significant better usability ratings than the other two UIs. During the experiment, participants were more engaged and stated that using the 3DUI is fun. The 2DUI could fulfill its purpose by being the best performing UI regarding speed and accuracy, but it was just slightly followed by the 3DUI. The Finger Count User Interface was rendered unsuitable for menu control regarding every aspect. Although the main idea of Finger Count was appreciated by the participants, it caused too much effort to even fulfill rather simple tasks and the tracking was not reliable enough to guarantee flawless interaction.This is just one example which shows us, that we are only at the beginning of what is possible in VR. It is possible to create a whole new world with its own laws, so the amount of still undiscovered User Interfaces is endless.

8 Future Work

8.1 Enhance UIs

The 2DUI and 3DUI had a dominant flaw in their sliders [see chapter 5.8]. Micro-movement caused many unwanted interactions when releasing the sliders' markers. So one could enhance this by making the markers harder to move the less velocity they possess. Because only when the markers' speed exceeds a certain value, we can be sure that the interaction is intentional.

8.2 Redo Study

The conducted user evaluation gave valuable insight and already showed some useful results. But still, eleven participants are not representative enough. To further verify our results, one could redo the same experiments with just more participants.
But it is also possible to get entirely new insights when altering the evaluation just a little and compare the results to our original findings. Here is a list of what can be changed:

- Redo the experiments with the enhanced UI
- Redo the experiments with a task that requires imprecise slider values
- Redo the experiments with a more precise input device than the Leap Motion
- Redo the experiments with a haptic feedback device, like the Dexmo exoskeleton glove [27]
- Redo the experiments and control the number of extended fingers during the interaction to get an even distribution

8.3 Remaining Ideas

During our whole work process, we collected many ideas which are worth mentioning and worth being observed.

Close on Look away

During our work, we mainly focused on using a UI, but we did not address what should happen before and afterward. What is a good way of opening the UI and what is a good way to close it again? Also, this matter is very context dependent. In a busy context, e.g. during a fight in a VR game, users do not have a lot of time to interact with a menu. For such hectic settings, we came up with the idea of closing the menu when users look away from it [see Figure 8.1]. For sure this mechanic would also need to be fine-tuned because closing it immediately after looking away is a bad idea. Here it is important to distinguish between intentionally looking away and unintentionally looking away. Using a short timer and threshold could help here.

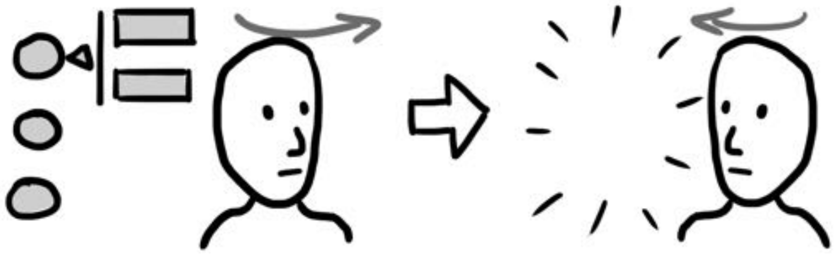


Figure 8.1: When looking away from the Menu, it will vanish [32]

Snake Menu

During development of the 2DUI, we noticed, that we do not need to remove the hand from a 2D Button to interact with another. We can just keep on intersecting from one button to another. This created an effect similar to scrolling a page which we enjoyed using. So we thought about a UI, where users have to transition from one button to another with their hand. Intersecting with a button will display its content until the final button is reached. The last button can open a submenu or a leaf of a tree, which the user just traversed. To open it, one needs to slide an element similar to the iPhone lock screen. To cancel the interaction with the menu, it is sufficient to stop intersecting the buttons. Because the movement through the menu can be similar to a snake, we call it the Snake Menu [see Figure 8.2].

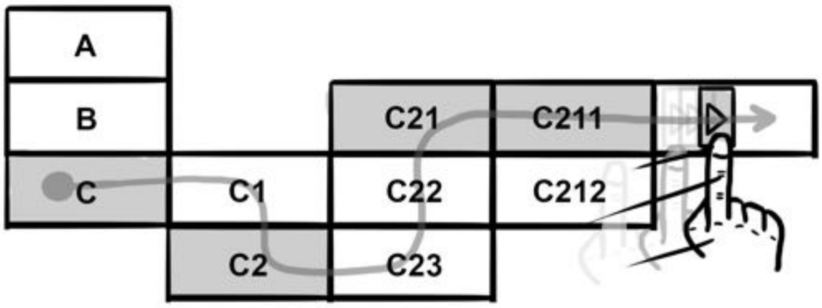


Figure 8.2: Touching any Element will open its subtree [32]

The Digit Input Dice

The Digit Input Dice is an idea which we actually implemented during early development. Unfortunately, it was unusable without haptic feedback and better screen resolution in the HMD. Fundamentally the Digit Input Dice is a ten face dice. From its center, a ray is constantly created to the user's eyes position, as long as the user looks at the dice. The dice's face, which gets intersected by the ray, is the currently selected face and the number which is printed on the face, is the currently selected number. To change the currently selected number, the user just needs to rotate the dice in his hand. To enter the number, the user just needs to cover the dice with his other hand shortly and thereby intersect the ray [see Figure 8.3]. Rotating the dice in one's hand is nearly impossible without haptic feedback, and with the currently available resolution of the HMDs, the dice needs to be pretty big, to read the number printed on it. So we do only recommend implementing this idea when resolutions better than the one the HTC Vive possesses are available. Also, a haptic feedback device is indispensable.

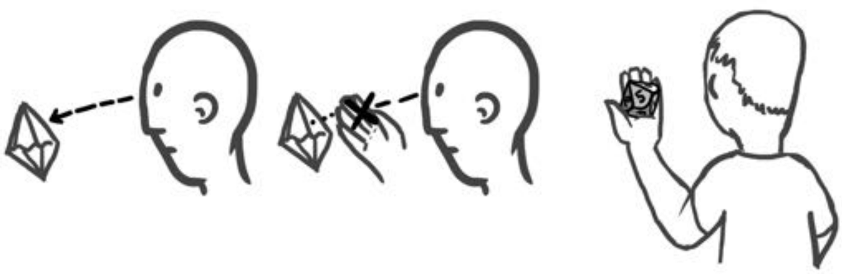


Figure 8.3: A ray from the Dice to the Head will mark a Surface and select a Number [32]

Ship Telegraph

The ship telegraph is normally used to control the speed of a ship. It is an excellent option to control speed while also automatically setting the direction. With its far longer lever, it could be a better alternative for precise slider input, then the slider we provided in our work [see Figure 8.4].

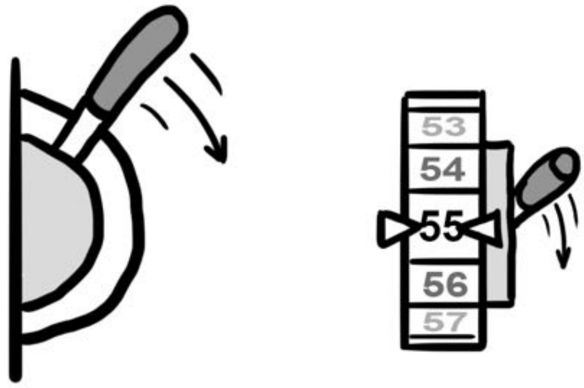


Figure 8.4: A concept similar to the one in ships [32]

Frankenstein Light Switch

This switch is famous from the Frankenstein film. It offers great affordance with its large lever and therefore it can be easily grabbed in VR. Combining it with our system that it gets pushed to the end of the other side when passing the center, makes it easy to use [see Figure 8.5].

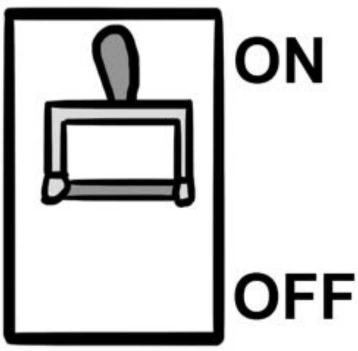


Figure 8.5: Famous from many Movies: The Frankenstein Light Switch [32]

Thumbs Up/Down Continuous Interaction for the FCUI

With this idea, one would break the consistency of the FCUI, because it is an interaction different from extending a certain number of fingers. Let us assume we want to interact with the Height Input Widget [see chapter 3.5]. Now one would use 4 FC Buttons to either increase or decrease the values. With the Thumbs up/down interaction, a user can continuously increase the values by simply showing a thumbs up and decrease them by showing a thumbs down [see Figure 8.6]. Of course, the user would need to select the value he wants to modify first. Showing a ten could still be used to finish the interaction with a value.

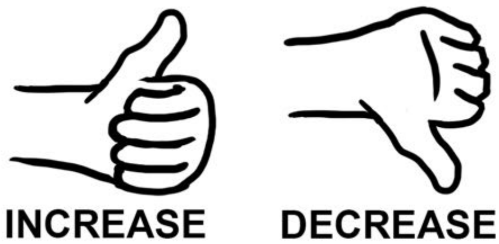


Figure 8.6: Increasing or decreasing the Values of Elements with simple Gestures [32]

Dynamic Thumbs Up/Down Interaction

This interaction is similar to the Thumbs Up/Down Continuous Interaction, which we just mentioned. But here the increasing and decreasing speed of the value is not constant. It is variable and depends on the angel of the thumbs up/down gesture [see Figure 8.7]. This interaction can not only be used to increase and decrease the values on a vertical axis. It can also be used to alter values on a horizontal axis and thereby this can be a very beneficial way to control a slider.

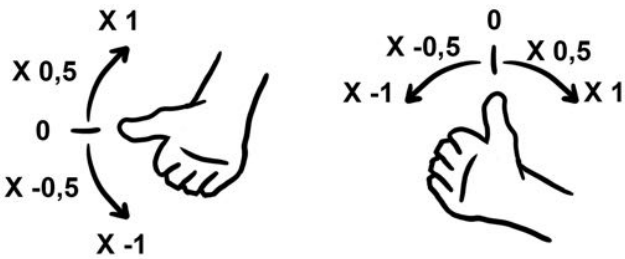


Figure 8.7: Increasing or decreasing the Values with far more precision [32]

Lamp Interface (The Pseudo Shadow Technique)

At one point of our work, we had to decide, how to illuminate the menu. We did exclude the UIs from the environment's lighting because we wanted to make clear, that the UI is not part of the environment. The UI was unlit and thereby always visible, no matter how dark the surrounding environment was. So we also enhanced visibility by this decision. Another approach could utilize a pseudo shadow. This technique was proposed and used by Chan et al. [23]. When a hand gets near a surface, the projected hand shadow will also get closer to the hand itself, which helps to estimate the distance to a surface by using pseudo feedback. To avoid confusion, where the shadow comes from and to establish a metaphor, we propose adding a light source to the UIs which matches the hand shadow [see Figure 8.8]. The usability of such a UI needs to be researched first and is worth comparing to our approach.

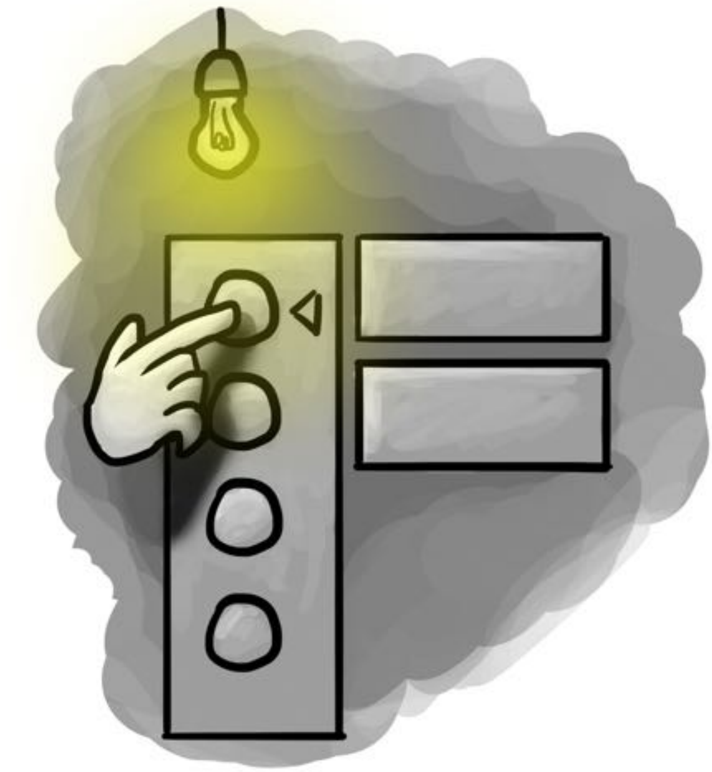


Figure 8.8: A Metaphor which supports the UI's Means [32]

8.4 Investigate further Fields of Application of Finger Count

Through our work, we showed that Finger Count is not suitable for menu navigation. Still, participants think it is very innovative and an interesting approach. So it would be nice to see where else it can be used. For example Kulshreshth & LaViola [36] researched the combination of many different input techniques and their effect on performance. In our opinion, it is also possible to use it in a hospital, to interact with any interfaces. One does not need to touch anything so it would be very hygienic. Maybe it can be integrated into shopping cards or used in VR Shopping applications as a quantifier. Because it can be used without seeing the actual interface, Finger Count could be a good option for blind people.

Acknowledgement

First of all, I am deeply grateful to my mother and my grandmother for always supporting me with my study. Also, I would like to express my sincerest appreciation to my future wife and her whole family for helping me overcome very challenging personal matters during the creation of this work. Moreover, I would like to thank Professor Dr. Antonio Krüger and every member of his chair for being so approachable and helpful. The whole team was a huge inspiration and motivation for me. Special thanks to my advisor Marco Speicher, who helped me a lot during the design and evaluation of the study. Additionally, I would like to show my greatest appreciation to Dr. Anusha Withana for reviewing this thesis. Also, a great thank you to my fellow student Vladislav Hnatovskiy for drawing and providing the concept arts for this thesis. And finally, I owe my deepest gratitude to my friends Omair Shahzad Bhatti and Philip Hell for all the times they helped me out when I was at my wit's end. Thanks to all of you.

Bibliography

[1] Axonvr - virtual reality you can feel. https://axonvr.com.

[2] Htc vive. https://www.vive.com/.

[3] Interaction engine: Early access beta. http://blog.leapmotion.com/introducing-interaction-engine-early-access-beta/.

[4] Leap motion. https://leapmotion.com.

[5] Leap motion core assets 4.1.6. https://developer.leapmotion.com/releases/core-assets-416.

[6] Leap motion frame system. https://developer.leapmotion.com/documentation/java/devguide/Leap_Frames.html.

[7] Leap motion's unity module for user interface input. http://blog.leapmotion.com/ui-input-module/.

[8] Unity configurabel joint. https://docs.unity3d.com/Manual/class-ConfigurableJoint.html.

[9] Unity events. https://docs.unity3d.com/ScriptReference/Events.UnityEvent.html.

[10] Unity hinge joint. https://docs.unity3d.com/Manual/class-HingeJoint.html.

[11] Unity spring joint. https://docs.unity3d.com/Manual/class-SpringJoint.html.

[12] Unity version 5.5.4. https://unity3d.com/de/unity/whats-new/unity-5.5.4.

[13] Unity3d. https://unity3d.com/de.

[14] Bowman, D. A., Kruijff, E., LaViola, J. J., & Poupyrev, I., (pp. 255–256). (2004a). *3D User Interfaces: Theory and Practice*. Redwood City, CA, USA: Addison Wesley Longman Publishing Co., Inc.

[15] Bowman, D. A., Kruijff, E., LaViola, J. J., & Poupyrev, I. (2004b). *3D User Interfaces: Theory and Practice*. Redwood City, CA, USA: Addison Wesley Longman Publishing Co., Inc.

[16] Bowman, D. A., Kruijff, E., LaViola, J. J., & Poupyrev, I., (pp. 315–322). (2004c). *3D User Interfaces: Theory and Practice*. Redwood City, CA, USA: Addison Wesley Longman Publishing Co., Inc.

[17] Bowman, D. A., Kruijff, E., LaViola, J. J., & Poupyrev, I., (pp. 147–148). (2004d). *3D User Interfaces: Theory and Practice*. Redwood City, CA, USA: Addison Wesley Longman Publishing Co., Inc.

[18] Bowman, D. A., Kruijff, E., LaViola, J. J., & Poupyrev, I., (pp. 141–143). (2004e). *3D User Interfaces: Theory and Practice*. Redwood City, CA, USA: Addison Wesley Longman Publishing Co., Inc.

[19] Bowman, D. A., Kruijff, E., LaViola, J. J., & Poupyrev, I., (pp. 303). (2004f). *3D User Interfaces: Theory and Practice*. Redwood City, CA, USA: Addison Wesley Longman Publishing Co., Inc.

[20] Butterworth, J., Davidson, A., Hench, S., & Olano, M. T. (1992). 3dm: A three dimensional modeler using a head-mounted display. In *Proceedings of the 1992 Symposium on Interactive 3D Graphics*, I3D '92 (pp. 135–138). New York, NY, USA: ACM.

[21] Butz, A. & Krüger, A. (2017). *Mensch-Maschine-Interaktion*. Walter de Gruyter GmbH & Co KG.

[22] Callahan, J., Hopkins, D., Weiser, M., & Shneiderman, B. (1988). An empirical comparison of pie vs. linear menus. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '88 (pp. 95–100). New York, NY, USA: ACM.

[23] Chan, L.-W., Kao, H.-S., Chen, M. Y., Lee, M.-S., Hsu, J., & Hung, Y.-P. (2010). Touching the void: Direct-touch interaction for intangible displays. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10 (pp. 2625–2634). New York, NY, USA: ACM.

[24] Ens, B. M., Finnegan, R., & Irani, P. P. (2014). The personal cockpit: A spatial interface for effective task switching on head-worn displays. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '14 (pp. 3171–3180). New York, NY, USA: ACM.

[25] Fitt, P. (1954). The information capacity of the human motor system in controlling the amplitude of movement. *Journal of experimental psychology*, 47(6), 381–391.

[26] Gianaros, P. J., Muth, E. R., Mordkoff, J. T., Levine, M. E., & Stern, R. M. (2001). A questionnaire for the assessment of the multiple dimensions of motion sickness. *Aviation, space, and environmental medicine*, 72(2), 115.

[27] Gu, X., Zhang, Y., Sun, W., Bian, Y., Zhou, D., & Kristensson, P. O. (2016). Dexmo: An inexpensive and lightweight mechanical exoskeleton for motion capture and force feedback in vr. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16 (pp. 1991–1995). New York, NY, USA: ACM.

[28] Ha, V., Wallace, J., Ziola, R., & Inkpen, K. (2006). My mde: Configuring virtual workspaces in multi-display environments. In *CHI '06 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '06 (pp. 1481–1486). New York, NY, USA: ACM.

[29] Hancock, P. A., Pepe, A. A., & Murphy, L. L. (2005). Hedonomics: The power of positive and pleasurable ergonomics. *Ergonomics in Design*, 13(1), 8–14.

[30] Hart, S. G. & Staveland, L. E. (1988). Development of nasa-tlx (task load index): Results of empirical and theoretical research. *Advances in psychology*, 52, 139–183.

[31] Hezel, P. J. & Veron, H. (1993). *Head mounted displays for virtual reality.* Technical report, MITRE CORP BEDFORD MA.

[32] Hnatovskiy, V. (2017). Commissional work for creating the concept arts in this thesis. In *Exploring Mid-air Interaction for Menu Control in Virtual Reality with regard to Task Performance and User's Preference.*

[33] Knight, J., (pp. 182–218). (1987). *Manual Control and Tracking in the Handbook of Human Factors and Ergonomics.* John Wiley & Sons: G.Salvendy (Ed.).

[34] Knöpfle, C. & Vöß, G. (2000). An intuitive vr user interface for design review. In *Proceedings of the Working Conference on Advanced Visual Interfaces*, AVI '00 (pp. 98–101). New York, NY, USA: ACM.

[35] Kulshreshth, A. & LaViola, Jr., J. J. (2014). Exploring the usefulness of finger-based 3d gesture menu selection. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '14 (pp. 1093–1102). New York, NY, USA: ACM.

[36] Kulshreshth, A. & LaViola, Jr., J. J. (2015). Exploring 3d user interface technologies for improving the gaming experience. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15 (pp. 125–134). New York, NY, USA: ACM.

[37] Laugwitz, B., Held, T., & Schrepp, M. (2008). Construction and evaluation of a user experience questionnaire. In *Proceedings of the 4th Symposium of the Workgroup Human-Computer Interaction and Usability Engineering of the Austrian Computer Society on HCI and Usability for Education and Work*, USAB '08 (pp. 63–76). Berlin, Heidelberg: Springer-Verlag.

[38] Liang, J. & Green, M. (1994). Jdcad: A highly interactive 3d modeling system. *Computers & graphics*, 18(4), 499–506.

[39] NASA, N. (1995). Std-3000. man systems integration standards. *National Aeronautics and Space Administration: Houston, USA.*

[40] Nathan, A. & Gao, S. (2016). Interactive displays: The next omnipresent technology [point of view]. *Proceedings of the IEEE*, 104(8), 1503–1507.

[41] Norman, D. A. (1986). Cognitive engineering. *User centered system design*, 31, 61.

[42] Poupyrev, I., Weghorst, S., Billinghurst, M., & Ichikawa, T. (1997). A framework and testbed for studying manipulation techniques for immersive vr. In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology*, VRST '97 (pp. 21–28). New York, NY, USA: ACM.

[43] Schrepp, M., Olschner, S., & Schubert, U. (2013). User experience questionnaire benchmark praxiserfahrungen zum einsatz im business-umfeld. *Usability Professionals*, 13, 348–353.

[44] Shupp, L., Ball, R., Yost, B., Booker, J., & North, C. (2006). Evaluation of viewport size and curvature of large, high-resolution displays. In *Proceedings of Graphics Interface 2006*, GI '06 (pp. 123–130). Toronto, Ont., Canada, Canada: Canadian Information Processing Society.

[45] Smith, T. & Smith, K. (1987). Feedback-control mechanisms of human behaviour. In *Handbook of human factors and ergonomics*, G.Salvendy (Ed.) (pp. 281).: John Wiley & Sons.

[46] Stoakley, R., Conway, M. J., & Pausch, R. (1995). Virtual reality on a wim: Interactive worlds in miniature. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '95 (pp. 265–272). New York, NY, USA: ACM Press/Addison-Wesley Publishing Co.

[47] Tan, D. S. & Czerwinski, M. (2003). Effects of visual separation and physical discontinuities when distributing information across multiple displays. In *Proc. Interact*, volume 3 (pp. 252–255).

[48] Thorpe, S., Fize, D., & Marlot, C. (1996). Speed of processing in the human visual system. *nature*, 381(6582), 520.