Designing experiences for virtual reality, in virtual reality - A design process evaluation

David Bergvik

David Bergvik
Spring 2017
Master Thesis, 30 hp
Supervisor: Ulrik Söderström
External Supervisors: Sonja Lakner
Examiner: Thomas Mejtoft
M.Sc. Interaction Technology and Design, 300 hp
1 Acknowledgements

There were many people that in some way helped me during this process, below is a list of people that I would like to give an extra thank you to:

Ulrik Söderström - My supervisor at Umeå University who helped me throughout this process with any inquiries.

Employees at North Kingdom - All you guys at the Stockholm office, thank you for including and inspiring me with your hard work, competence and feedback during this thesis and internship.

Sonja Lakner - My mentor at North Kingdom who guided and supported me, as well as including me in external projects at NK. Thanks!

Simon Johansson, Maja Andersson and Freja Wieslander - My peer-review group, without your feedback and motivation this thesis would not be all that it is today.

My parents, my sister and Johanna Bäckström - Thank you for all of your love and emotional support, all day every day.
2 Abstract

Creating immersive experiences for virtual reality (VR) presents new design opportunities and challenges that do not appear when creating experiences on a screen. Creating prototypes and exploring concepts in VR is today limited to professionals with previous knowledge in 3D application development, and testing 3D experiences requires the usage of an Head-Mounted Display (HMD), which forces professionals to switch medium from the computer to an HMD. With new advances in this field, there have to be new solutions to these challenges. The goal of this thesis is to explore how VR technology can be utilized in the experience design process for VR. This is achieved through a literature study and conducting expert interviews, followed by a hardware evaluation of different HMDs and concept creation using rapid prototyping. From the interviews, a number of issues could be identified that correlates with the research from the literature study. Based on these findings, two phases were identified as suitable for further improvements; Concept prototyping and testing/tweaking of a created experience. Lo-fi and hi-fi prototypes of a virtual design tool were developed for HTC Vive and Google Daydream, which were selected based on the hardware evaluation. The prototypes are designed and developed, then tested using a Wizard of Oz approach. The purpose of the prototypes is to solve some of the issues when designing immersive experiences for HMDs in the suitable experience design phases that were identified by analyzing the interview results. An interactive testing suite for HTC Vive was developed for testing and evaluation of the final prototype, to verify the validity of the concept. Using Virtual Reality as a medium for designing virtual experiences is a promising way of solving current issues within this technological field that are identified in this thesis. Tools for object creation and manipulation will aid professionals when exploring new concepts as well as editing and testing existing immersive experiences. Furthermore, using a Wizard of Oz approach to test VR prototypes significantly improves the prototype quality without compromising the user experience in this medium.
3 Introduction

Recent advances in the field of Virtual Reality (VR) using Head-Mounted Displays (HMD) has presented a new medium for immersive experiences. Designing for this medium does have its challenges; As the medium for the experience, Head Mounted Displays (HMD), differs from the medium on which it is created (desktop PC), designers and developers have to adapt new design processes and pipeline for creating and manipulating 3D environments in VR. These professionals need a new toolset to effectively work with this new medium as it’s emerging into an entire industry. There have been a lot of research within similar fields of 3D manipulation with a Head Mounted Display (HMD) [1] [2] [3] [4], but recent technological strides demands new efforts to translate earlier knowledge toward new application of these professions and what tools are needed.

Since the introduction of computers more than five decades ago, a great number of new professions and companies have emerged as this new technology creates more jobs for society. One of the applications that harness the power of computer systems are computer graphics, a term coined by William Fetter in 1960 [5]. In the 1990’s the graphics took the leap into 3D modeling for the consumer market, and with the increased processing power in today’s computers, a user can create complex 3D environments on their personal laptops. 3D designers create and manipulates 3D environments using powerful computers, high-resolution screens and a number of input tools like pens, pads and keyboards.

VR is a technology that’s been around for decades and has in the last few years developed into a whole new set of entertainment, and keeps getting more traction each year [6]. VR is a concept in which the user is placed in a virtual reality using a headset or other hardware. This is the type of VR we see today, but the concept was used as far back as the 1800s, with the panoramic paintings that stretch 360 degrees around the spectator. In recent years, the variety of VR headsets has exploded and technology became during 2016 a billion-dollar-industry and is expected to continue increasing in the coming years [7]. This increase created a tremendous increase of available application for VR and there are as of 2016 hundreds of companies working with developing experiences for this new medium.

The process of creating 3D experiences in VR comes with challenges and issues not previously seen in this field. Is there a way to utilize the opportunities of this

---

3http://www.marxentlabs.com/job/virtual-reality-jobs/
4https://www.theguardian.com/business/2015/aug/17/technology-created-more-jobs-than-destroyed-140-years-data-census
6http://venturebeat.com/2017/03/08/vr-companies-grew-40-percent-in-2016/
new medium and in that case, how could that redefine the design process? Is this preferable to the current way of working? Let’s find out!

3.1 North Kingdom

This thesis will be carried out together with the company North Kingdom. The company defines itself as an "experience design company", founded in Skellefteå and has been a public company since 2003. Today, the offices of North Kingdom are in Skellefteå, Stockholm and Los Angeles. Their vision is

We Believe That new value can be created wherever people, business, and technology collide. We help our clients harness That value through the creation of experiences, products, and services That play a meaningful role in people’s lives. Through human-centered design, we make the complex simple and relatable, no matter what medium or platform.

This thesis will run in parallel with North Kingdom's current VR projects. Their design process and the pipeline will act as the scope of this thesis.

3.2 Experience Design

Definitions of the term "Experience Design" is widely spread out throughout design communities, pinpointing different parts of the design spectrum. The main concept, however, is explained by Marzano as a process of focusing on the quality of the user experience and shifting the focus from expanding sets of features towards solutions that are tailored to the current cultural state of society [8]. Furthermore Marzano explains how companies like Philips Design focuses on the complete experience that a user has with their brand. This extends from the first impression, through the usage lifecycle, and to the overall experience. Marzano presents a design process created by Philips Design described as "The Experience Design Lifecycle", which consist of three main activity phases; Envisionment, Experience Concepts and Experience Centre. These concepts will be explained further in the following section.

**Envisionment:** This stage is where technical developers and designers collaborate to understand the possibilities and effects of different design elements, isolated from their intended context. This is carried out using rapid prototyping (more details in section 6.5) and is explored with isolated elements and as a collaboration between different elements.

**Experience Concepts:** Pinpointing cultural trends and human needs is essential in this stage when initial concept ideas are being explored. The core aspects defined as the foundation of this phase are People, Space and Enablers over time. This provides a context which is used to create a unique and tailored experience.

- The people aspect focuses on the individuality of people and the uniqueness of their individual experience. This re-aligns the focus of the design, from technological capabilities to experiences of the users.

---


• Space symbolizes the context of which the concept or product is perceived and used. The focus of this aspect is to utilize and account for space and how it affects the user.

• Enablers are the technological advancements as well as design enablers that connect the system to the user. The key to success in this aspect lays in the combination of these tools in reference to the user’s needs and capabilities.

• All of these three have to be adapted to the presence of time and how that affects the attention of the user. The result from this stage is experiential prototypes that embody the core features and elements of the concepts, which is used as a demonstration of the ideas as well as a subject for testing and evaluation.

Experience Centre: At this stage, the concept is brought into the wild, the real world, through the creation of prototypes that can withstand proper usage and placement in its targeted environment. This provides insights of effects, usage and acceptance of real users. The cost and effort of creating these prototypes are much higher than the experimental prototypes created in the previous step in order to give accurate responses from users. This phase also implies that a ‘parent-child’ approach should be considered if a new technology is being introduced to the environment. This means that the new emerging technology is supported by an established technology as a foundation.

3.3 Purpose

As companies might shift their focus towards creating experiences in virtual reality, their design process has to adapt according to the features of this new medium. As a part of that, this thesis will evaluate the opportunity of a VR tool for manipulating 3D environments while immersed in VR, into their new VR design process. The purpose of this is to elevate the experience when creating and using the end product, as designing 3D environments and experiences is highly dependent on the actual perspectives and positions of the end user. When designing these environments on a screen some aspects of the design becomes hard to predict when seen on a regular screen.

3.4 Objective

Allowing the designer and developer to manipulate the environment of the experience when immersed, might solve issues explained in section 3.3 and create a more intuitive user experience during creation and manipulation of objects in the application. By creating a tool that can allow this type of workflow, the goal is for the user to make instant changes to the world without having to leave it. An evaluation of this tool will also be conducted, if it can be used for rapid prototyping and bridge the knowledge gap between team members by letting non-developers edit the prototype their ideas. Here follows the main goals of this thesis:

• Research possibilities and drawbacks of using VR as the medium for creation
and manipulation.

- Study how VR projects are carried out today through interview, then identify opportunities where immersion is beneficial
- Design VR interface prototypes that are user-friendly and task specific from the results of the research and study
- Build functional prototype based on the final design
- Prototypes will be evaluated with user tests

3.5 Limitations

This thesis will be based on the current technology and software on the market. The concepts and prototypes created will be restricted to the possibilities of what can be implemented into a working system today. The scope of what will be investigated is restricted to the field of VR experience and experience design along with its implementations. The targeted user-base will be limited to working professionals within the field of tech and experience design. The hardware HMD selection will be limited to the available hardware at the Stockholm office of North Kingdom.

3.6 Terminology

Common terms and their explanations are presented in this section:

- Field of View (FOV): What a user can see without turning head or body
- Head-Mounted Display (HMD): A display device is worn on the user’s head with a small screen in front of one or each eye
- Virtual Reality (VR): A digital world where the user gets immersed. This cannot be seen in the real world
- Virtual Environment (VE): Replicates the real world and the user’s presence and interactions inside it.
- 360° video: A video feed that can be seen from all angles, not only a specific camera angle.
- Degrees Of Freedom (DOF): The number of parameters in which an object or a system may vary independently.
- Application Programming Interface (API): A set of defined methods of communication between several types of software.
4 Related Work

There has been a lot of research in the field of Virtual Reality and manipulation in a Virtual Environment (VE) which was most recent summarized by Jankowski in 2015 [9]. Further reading about specific techniques and tools can be found in Section 5.4. There have been a number of studies and implementations of systems that manipulate 3D environments using an HMD. The first major stride within this was in a study Butterworth in 1992 where a 3D surface modeling program was developed [10]. This system was created using the same principles as CAD\(^1\) and similar software withing 3D object creation and manipulation. Since then there has been numerous other studies and applications within 3D object creation with an HMD [2] [11] [12].

Apart from the objects themselves, there has been some studies about using an HMD to manipulate 3D objects in relation to a VE. This approach is not as focused on 3D modeling as much as positioning and scaling of pre-fabricated objects in a 3D environment. In a study by Stoakley in 1995 a system called World in Miniature (WIM) was created [3]. The premise of this system is to display a miniature version of the VE as well as the full-scale one. This allows the user to perform delicate manipulations at full scale, and bigger more fundamental changes on the miniature version. The user is holding a clipboard in one hand which represents the miniature world, and an interactive ball in the other for selecting and manipulating in 3D space. The same year, Mine introduced the "Immersive Simulation Animation And Construction" system (ISAAC) [4]. This system is primarily designed for scene composition and interactive construction of virtual worlds. It tries to take advantage of the possibilities in the virtual world but keeps the interaction scheme similar to the real world for easier adaptation to less experienced users. ISAAC uses multiple types of tools for object manipulation, such as ray-cast and direct manipulation (see Section 5.4) and multiple tools for the user to travel. This system is a showcase more then an application for real-world usage, as it demonstrates many of the possibilities with object and scene manipulation using an HMD but does not limit interaction tools based on a specific purpose or target-group. A concept with an intended purpose and demographic in mind was however presented by Deering, again in 1995 [13]. The tool, HoloSketch, was created as a "what you see is what you get" virtual reality concept for real-time creation and manipulation of 3D objects and assemblies. It is designed for users with little to no programming knowledge to be able to create complex 3D objects and simple animations of an object. The display unit is based on the Virtual Holographic Workstation [14], which uses an external CRT monitor and head-tracked field sequential stereo shutter glasses. The virtual objects are manipulated with a 6 DOF wand/controller. The main user interface (UI) is a circular menu with diagonal lines to separate the different options(pie-menu), visualized in Figure 1. This UI is visible upon a button-press which hides the VE and displays the menu. This tool is

\(^{1}\)http://www.autodesk.com/solutions/3d-cad-software
There have been some studies about merging a regular workstation for 3D manipulation with an HMD [15] [1]. Bowman et al. researched and developed with an objective similar to the objective of this thesis back in 1997 [1]. The authors investigated how to combine a regular work setup for 3D development with an HMD to remove the repeated transition between the two. Their Augmented Reality (AR) system is based on a real workstation which consists of a computer together with screen, keyboard and mouse. This is combined with a Projective Head Display (PHMD) to create an environment where the user can see projected 3D objects and still be able to use the workstation as usual.

This study will combine these previous findings and concretize how the usage of these tools can be adapted to the VR hardware available today for experience design projects.
5 Theoretical Framework

This section presents a theoretical framework based on the literature study. The framework contains six main parts: 5.1 Head Mounted Displays, 5.2 User Centered Design, 5.3 Interaction Issues, 5.4 Interaction Tools in VR, 5.5 Interface Design and 5.6 Best Practice Evaluation.

5.1 Head-mounted Displays

The most common way of experiencing an immersive virtual environment is with a Head-mounted Display (HMD). This device is placed on the head of the user with separate views, one for each eye. The displays produce slightly different images which creates a stereoscopic view when perceived through the HMD. The experience is called stereopsis [16]. This is achieved in two different ways as of the current HMDs available in March 2017; Two different displays integrated into the headset that produces different images or one removable display that displays two separate images on the same screen. The second approach utilizes the steady rise of high performance smartphones with high resolution displays into the mobile market[^1], as a smartphone can be used as a display and central processing unit for these HMDs. HMDs can be abstracted based on their functionality, summarized by Bierbaum et al in their study about creation of virtual application using HMDs [17]. Based on HMDs available today, three classes of HMDs can be obtained using this way of abstraction based on interaction possibilities and performance:

- **Low-level** HMDs has a removable display (mobile device) with internal tracking. Interactions are limited to the headset, no external interaction tools.

- **Mid-level** HMDs has a removable display (mobile device) with internal tracking. Interactions extends to one remote interaction tool (controller) with internal tracking functionality.

- **Hi-level** HMDs has two (2) internal displays, run on external machine. Interactions includes two controllers. Headset and controllers are tracked internally and with external sensors.

5.2 User-Centered Design

According to Stanney et al. it is crucial to consider the human factors when designing successful applications in virtual reality environments. When the end user has little to no technical knowledge about the processes going on within the designed system or application, accounting for this, it is a valuable tool for a good user experience. [18]

[^1]: https://deviceatlas.com/blog/16-mobile-market-statistics-you-should-know-2016
5.2.1 Cognitive Load

The human brain has a limited load capacity, which is where currently used information is stored [19]. This load capacity can be compared to the working memory of a computer. Cognitive load is explained by Sweller as the amount of mental usage that is used in the working memory [20]. When trying to solve a problem with the desired goal state, external uncertainties increases the cognitive load as the information has to be stored in the working memory. Sweller states that instructional design can reduce the amount of cognitive load, which increases the learnability within the task. Another tool that is used to reduce this load is by documenting information between steps in the problem-solving process in order to free that space from the working memory [20]. Without this tool, the working memory has to keep that information as well as the current state.

5.2.2 Flow

Flow is a term for describing a state of focus within creative work and was coined by Csikszentmihalyi in 1975 by studying playfulness, creativity and the characteristics of artists for years. [21] The author claims that by studying pleasure we can move away from tedious tasks in a professional environment. Flow is a powerful way of exposing creativity when solving a problem or creating something, and the building blocks of flow is presented by Csikszentmihalyi as follows: [22]

- There are clear goals every step of the way
- There is immediate feedback to one’s actions
- There is a balance between challenges and skills
- Action and awareness are merged
- Distractions are excluded from consciousness
- There is no worry of failure
- Self-consciousness disappears
- The sense of time becomes distorted
- The activity becomes autotelic

5.3 Interactions issues in Virtual Reality

Virtual Reality brings a new medium with new possibilities but is not without drawbacks. There are limitations that has to be taken into account when working within the bounds of the medium. The purpose of this section is to highlight the pain points when designing for VR in order to achieve a better user experience.

5.3.1 Generality issues

The selection and design of an interaction for an application is hard to apply to a specific problem or context. From their extensive research about 3D interfaces,
Bowman et al. [23] summarize four ways that the majority of interaction techniques exhibit generality:

1. Application- and domain-generality: The technique was not designed with any particular application or application domain in mind, but rather was designed to work with any application.

2. Task-generality: The technique was designed to work in any task situation, rather than being designed to target a specific type of task. For example, the design of a pointing technique (like a ray-cast, see Section 5.5.1) does not take into account the size of the objects to be selected and becomes very difficult to use with very small objects [24].

3. Device-generality: The technique was designed without consideration for the particular input and display devices that would be used with the technique. Often techniques are implemented and evaluated using particular devices, but the characteristics of those devices are not considered as part of the design process. For example, the HOMER technique [25] is assumed to work with any six DOF input device and any display device, but all of the evaluations of this technique have used a wand-like input device and a head-mounted display (HMD).

4. User-generality: The technique was not designed with any particular group of users or user characteristics in mind, but rather was designed to work for a "typical" user.

5.3.2 Occlusion problem

A problem with interactions in VR, that has a very small significance on screen-based UI, is occlusion since the user interacts and moves in a VE with 3D objects, the possibility of objects blocking each other [26]. To solve this the user can move around in the virtual space and try to find an angle that occludes the object, or use a selection tool (see Section 5.4) that can be bent around the first object or pass through it. Mossel offers in her study a different solution, where the user can "slice the environment and hide it in order to get access to the desired object [27]. This method was preferable from the standard method which is to move and find a better angle.

5.3.3 Human body limitations

One of the main differences with interacting in a virtual environment compared to a screen interface is the importance of the entire human body (not just hands), which brings both opportunities and limitations. This section will shine a light on some major limitations that are important to keep in mind when working with VR interfaces and interactions.

Reach

Physical reach is also a big problem when interacting in a virtual environment. It limits the interaction space to the length of the user’s body (most often arms) if the interaction technique is direct manipulation (see Section 5.5.1).
**Field of view**

When working with interfaces on a screen, the designer and the system knows what is visible to the user at any point in time. When diving into VR however, it works the other way around. The user has full control of what is visible at any time, which has to be accounted for. The origin of this is the same that we experience every day in our daily lives, we have information all around us but our vision is restricted to our field of view. The field of view of the user (in VR and the real world, just by moving their eyes) is about 94 degrees in a cone shape (circular vision). Past 77 degrees is considered peripheral vision and is not focusable without turning. The user can rotate their head 30 degrees without any constraint, and max out at 55 degrees. Looking up is 20 degrees as comfortable and 60 as max. Looking down is comfortable to about 12 degrees with a max of 40 degrees. (figure 2b. By combining all of this we get three directional zones for content (figure 2), as explained by Alger in his study about UX for VR [15]:

- **Main Content zone**: $0 - 77^\circ$ This is the field of view for the user without strain. Usually where the primary UI goes (figure 2a.a).

- **Peripheral zone**: $77^\circ - 102^\circ$ The user have to physically strain and turn their head to see this zone. It does still prove suitable for UI of less regularly use (figure 2a.b).

- **Curiosity zone**: $102^\circ - 158^\circ$ This zone is only visible when the user rotates their body (figure 2a.c).

There are also 3 depth zones, which represents how the object is perceived. Those are:

- **No content zone**: $0 - 50$ cm. Alger explains that this is because objects that appear closer than that will cause the user’s eyes to cross (figure 2a.d).

- **Primary depth perception zone**: $50$cm - $20$m. Within these distances, the user can perceive objects with depth.

- **Beyond the horizon**: $20$m - MAX. Objects at this distance loses their depth perspective.

By combining three zones that represent where different kinds of content should appear, Comfort-perifial-curiosity zones can be visualized as well. [15]

### 5.3.4 Physical Space

The journey to a virtual environment using a portable headset does not include a vast infinite empty physical space to move around in. This causes problems when users are immmersed as they cannot see the physical objects in the real world which can and have caused injuries [28, 29]. Some features to the high-end segment to prevent this issue include a visible barrier to inform the user of where to move around freely. This is, however, something that has to be manually setup before use. [30]
5.4 Interaction Tools in VR

There is a lot of parameters in a VR to take into account when deciding on a selection tool and technique, which is explained by the great number of studies that have been conducted in this field of VR [31–34]. Some of the tools and techniques that exist for VE will be explained.

5.4.1 Direct Manipulation

One of the best and most intuitive ways of interacting in VR are using direct manipulation [35]. This technique mimics the way that humans interact and manipulate objects in the real world; Grabbing with our hands. There have been studies with variations of this technique which have proved that it performs great as an intuitive tool for interacting in VR [33,36]. These studies also conclude that direct manipulation is preferable when the target users are novices at using VR. Direct manipulation can also be used for traveling within the VE by using Scale-World grab, which was created by Mine [32]. This allows the user to move around by grabbing inside the VE and moving hands in the opposite direction of where the desired destination is. For example: If the user wants to go forward then they’ll extend their arms forward, grab, then pull their hands towards their body. Furthermore, Mine explains how the Virtual Hand (direct manipulation) can be combined with a raycast (from section 5.5.1) to provide delicate object manipulation and also reach occluded objects. This is explained as a technique shift between the two that occurs when the hand/controller appears/disappears from the user’s field of view.

5.4.2 Raycast techniques

The raycast technique is similar to the user having a laser pointer in their hand, where objects are targeted by pointing at them. Studies have concluded that in an environment with a sparse selection of objects with a volume that is not too small,
using a raycast is fast and reliable [37]. When some of these parameters change, the raycast tool with a "laser-pointer" technique experiences more issues. In an environment that contains a lot of objects in a small space, the error rate when trying to select an object rises [38]. This factor is multiplied when movement is added to the object (typical for games).

Multiple studies argue that there are better ways to perform object selection in a more complex and dynamic environment by tweaking this concept [25,39]. By using techniques that are designed for dynamic and cluttered environments the speed and error rate can be reduced. Two of these techniques are 'zoom' and 'expand'. When using 'zoom' the surrounding area of the selected object is "view through a magnifier" upon object selection to simplify the selection. Using "expand" upon selection resizes the object and adjacent objects to a larger size, for easier selection.

This tool is not without setbacks though. One problem that arises is the "lever-arm" problem which describes how an object that’s selected with a raycast has limited number of possible manipulation actions [40]. Another big problem with ray-cast is trembling of the hand and twitches that occur when a user tries to select an option. This has been given the name "Heisenberg effect" and is the cause of new interaction issues that arrives with this new medium [41]:

- User dissatisfaction due to increased error rates.
- Discomfort due to the duration of corrective movements, which in the absence of physical support require an additional physical effort.
- Unconfidence on which object will be selected after triggering the confirmation.

5.4.3 Selectors on objects

Strauss et al. present a selector in his course notes [42] as visualization of actions or operation to an object, in proximity to said object. The authors continues by listing benefits from utilizing selectors as listed below:

- A number of different controls available to the user at all time
- Allows the user to isolate manipulation in 3D-space. This can limit the object to only be manipulated along the X-axis even though the user input is along all three axis’.
- It displays the object that is manipulated and what operations are possible on an object at any time.
- The user’s attention stay on the object as the tools are in direct proximity to the active object.

5.4.4 Environment Positioning

Being able to position yourself within a virtual environment is fundamental in order to interact with it and is important to investigate [43]. Even if the physical space around the user is confined (see section 5.3.3 for more information), the virtual space can be infinite. In order to harness the power of an immense virtual environment, some kind of positioning tool or traveling technique can be implemented. This
will allow the user to interact with a distant object through direct manipulation, get a different perspective of the virtual world and more as explained by Robinett and Holloway [44]. The authors present in the study the three actions that can be implemented into a virtual environment and are fundamental for positioning:

- **Translate** Move object, teleport/transport the user
- **Rotate** Grab and turn an object, tilt the entire world
- **Scale** Scale an object, expand/shrink the world/user

A positioning system can be created by selecting and combining tools and techniques in Section 5.4.

### 5.5 Interface Design

Interfaces in a virtual environment come with a new set of challenges when compared to a traditional interface designed for a screen, these will be discussed in this section. When designing an interface it’s important to evaluate the speed of selection. Fitts’ law is a fundamental and proven way to evaluate pointing to real-world objects by measuring the distant to the object and its size [45].

This is however based on a real-world scenario, which does not translate into a virtual environment, where the user need a tool to interact with objects. Despite these differences, Fitts’ law can be applied to pointing in a virtual environment using the following formula [46]

\[ ID = \log(2) \times (2 \times \frac{D}{W}) \]

where \( ID \) is the index of difficulty, \( D \) is the distance from starting point to the middle of the correct target, \( W \) is the width of the target (calculated on the axis where the pointer will travel).

### 5.5.1 Interface Design for Virtual Reality

When designing UI for VR there are new challenges and choices to make in order to please the user and keep the functionality of the UI. The interface of a VR system is explained in Sherman’s extensive book about VR [47] as:

The access point through the boundary between the recipient and the virtual world is the user interface

Sherman explains how virtual worlds are designed for a specific medium and interface and later adapted to facilitate for different user interfaces. This often brings a lower quality to the adapted system according to Sherman who later concludes that the critical part of the process lies in selecting a suiting medium for the intended goal and content [47].

**Placement and Head-Up Display**

In order to present information to the user in VR approaches like in-world UI’s or Heads-Up-Displays (HUD) can be appended. An in-world UI acts like a physical
object in the VE and has a world-related position. An example of an in-world UI is a selector which has a position relative to the selected object. Interactions with this type of UI can be achieved using techniques like direct manipulation and ray-casts. A HUD is an interface that is attached to the user’s field of view and does not have a position related to the VE. HUD usage in VR applications consists of displaying information about the user, virtual object or the environment [47].

5.6 Best Practice Evaluation

Since the release of both the Oculus Rift\textsuperscript{2} and HTC Vive\textsuperscript{3} which both have 6 Degrees Of Freedom (DOF) controllers, the number of applications focusing on complex interactions and manipulation has really exploded\textsuperscript{4}. Regular web applications as well as HTC Vive applications that manipulate 3D objects in a 3D environment was analyzed and evaluated, from with greater insights into popular usage and implementations was gathered.

5.6.1 Storyboard VR

The creative team at Artefact \textsuperscript{5} created a prototyping tool for VR called Storyboard VR\textsuperscript{6}. This tool gives the user the option to create high fidelity\textsuperscript{(see section 6.5.2)} VR prototypes while immerged with a HMD. Objects in the scene are uploaded as high-resolution images and placed in the VE. The essential interactions of this system are explained in this section.

Tools

To interact with this system the primary controller is equipped with a ray-casting technique (see section 5.5.1) for selection within the VE. Most tools in this system can be found on a swipe-based menu, attached to the secondary controller\textsuperscript{3}. There are some object-specific selectors (see section 5.5.1) attached to each object, these are visible when the object is selected by the user. It is not possible to teleport or in other ways travel in the application, but there are a timeline/storyline where different ‘frames’ can be accessed.

- **Creating objects** In order to create an object within the world, the user can select from imported images or default 3D shapes from the ‘Create’ part of the menu. An object that has been created can later be duplicated by selecting ‘Duplicate’ on the same menu and then selecting the object\textsuperscript{3a}.

- **Selecting and deleting objects** Objects are selected by pointing the ray from the primary controller on the and clicking the trigger. An object is deleted by selecting the object and selecting remove on the menu\textsuperscript{3b}.

\textsuperscript{2}https://www.oculus.com/rift/
\textsuperscript{3}https://www.vive.com/eu/product/
\textsuperscript{5}https://www.artefactgroup.com/
\textsuperscript{6}https://www.artefactgroup.com/work/storyboard-vr/
• **Scale and positioning** Changing the position of an object is achieved by pointing the ray at the object, holding the trigger while moving it into position with the ray, then releasing the trigger. The scale of an object is manipulated by swiping up or down on the touchpad on top of the primary controller.

5.6.2 Tilt Brush

One of the applications that were released with HTC Vive is the praised application "Tilt Brush" by Google \(^7\) where the user can use the existing hand controllers as paintbrushes and paint in 3D space. This application highlights many of the interactions that are essential when manipulating a VE. Some of these interactions are explained further in this section.

• **Creating objects** Creating a new painting stroke is the core of the application, and can as such be accessed directly with primary triggers on both controllers.

• **Selecting and deleting objects** Strokes are targeted by placing a spherical tool (attached to your brush) around the stroke and pressing the same trigger that is used for drawing. The tool is activated by selection on the secondary brush menu.

• **Parameters and setting** Tilt Brush mimics the way that painters paint in many ways, most prominent when selecting and changing parameters. The primary controller acts like a paint brush and the secondary controller acts as a painters palette. The most common and most used tools for what will be painted (brush-size, undo, deleting strokes etc) are accessed through the primary brush (controller). More complex modifications (brush type, colors etc) are accessed from the secondary brush by activating a menu connected to that brush (controller) and selecting with the other brush.

• **Scale and positioning** One of the biggest advantages of working in a VE is that you are not bound to the restrictions of your physical relationship to the

---

\(^7\)http://store.steampowered.com/app/327140/?l=english

\(^8\)https://www.tiltbrush.com/
objects that you are working with. By using both controllers and moving them away from each other, the environment grows and the users scale decreases. The user scale can be enlarged by moving the controllers closer together. By using a raycast (see Section 5.4) the user can teleport around in the VE.

### 5.6.3 Vizor

Vizor\(^9\) is a web-based application that allows the user to create VR experiences and applications in a web-browser. The application is based on WebVR \(^10\) and is Object Oriented (OO), where each object in a scene has properties in the form of other objects. Some of these properties (positioning, scale and rotation) can be accessed and edited in "Build" mode (see Figure 5a) while more complex properties and inner object are accessed through the "Program" mode (see Figure 5b). This approach allows the user to get a holistic view of the entire scene and all objects as well as their properties and relations to each other. Instead of a list hierarchy with all of the objects, the scene object is the central hub, connected to all objects in it (except for inner objects that are connected through a parent object). One of these connections is shown in Figure 6a. The approach to dividing the process into two parts is also present in the form of objects and patches, these are displayed in a list (Figure 6b). Objects are regular 3D objects that can be added to the scene, patches are properties that can be added to these objects.

---

\(^9\)https://vizor.io/
\(^10\)https://webvr.info/
(a) Build mode. Objects can be positioned and scaled with mouse actions.

(b) Program mode. Properties and objects can be connected and edited using mouse.

**Figure 5:** Different modes of web application Vizor

(a) A connection between the scene and the main camera

(b) Program mode. Properties and objects can be connected and edited using mouse.

**Figure 6:** Different elements in Vizor
6 Methods

The methodology used in this thesis is presented in this section, that includes a literature review, best practice evaluation, internal and expert interviews, an analysis and prototypes.

6.1 Literature Study

In order to support the design and implementation in VR, a literature review was conducted on a variety of fields within the VR spectrum. The study included topics about VR basics and history, interaction tools and techniques within VR and what kind of applications was released up until February 2017. A brief study of different hardware specifications were also conducted to attain knowledge about the possibilities within each medium/HMD. The selection of hardware was based on the existing software support for implementation and the design processes that were discovered during interviews (see Section 7.2). The study will for the same reason include work ergonomics and explore what possibilities, restrictions and impact this would have on a VR working environment. Some literature about prototyping in VR was also investigated (Lo- and Hi-fi prototyping for VR) as prototyping was later used as a technique for designing based on the results of the study, which can be found in section 5. The literature was collected from scientific articles, books, periodicals and recognized credible news and blogs.

6.2 Best Practice Evaluation

Popular applications and tools for creating and manipulating a VE was analyzed as a part of the theoretical framework. This was done to serve as a benchmark and to gain further insights into how solutions are implemented today using different mediums and tools. A total of three applications where analyzed and evaluated; Storyboard VR\(^1\), Tilt Brush\(^2\) and Vizor\(^3\).

6.3 Interviews

By conducting interviews with professionals within VR development, design and projects, information about what is lacking today and what opportunities exists was extracted. As these interviews were focused on the individual perspectives, ideas and

\(^1\)https://www.artefactgroup.com/work/storyboard-vr/
\(^2\)https://www.tiltbrush.com/
\(^3\)https://vizor.io/
experiences about the process of creating a VR experience, in-depth interviews were conducted [48]. The results can be found in section 7.2. The interview where structured after the "Three Boxes" method, which consists of an introduction, body and conclusion [49]. The entirety of the interview structures can be found in Appendix A. The interviewee answers were analyzed using ‘Grounded Theory Methodology’, which is presented by Strauss and Corbin [50] as explicitly

generating theory and doing social research [as] two parts of the same process.

6.3.1 Internal Interviews

In order to understand the current design process and general work issues in VR projects at North Kingdom, internal interviews were conducted. These interviews contributed to the selection of parts within the design process that will be the foundation of the result of this thesis. Among the interviewees where Robert Lindström, a Swedish Art Director and Designer, Head of Design Innovation, Owner and Co-founder of the company. David Ljunghill, a UX lead were also interviewed along with Unity Developer Oskar Eriksson.

6.3.2 Expert Interviews

As companies in the VR industry are developing a specific pipeline and design process for their VR projects, expert interviews were conducted to get a broad sense of the current practices. The main focus was to get a better understanding of how pipelines and design processes are carried out in VR projects. It also serves as an opportunity to investigate if the internal processes at North Kingdom were missing some fundamental parts in their design process for VR projects. In order to get the insights from within other fields and applications of VR the interviews were conducted with professionals from different parts of the industry. One interview was conducted with Elin Sjöström, a UX designer from a VR company called byBrick, and another with Joel Ring who is one of the founders of the VR-hub called VR Sverige.

6.4 Analysis

After the literature study and the interviews were conducted, an analysis of the problems and possibilities were conducted. This phase defines which core concepts that should be selected for the prototype, by correlating the possibilities from earlier research (section 5) and current solutions (section 5.6) with the issues and expectations in the industry today (section 7.2).

⁴http://www.bybrick.se/
⁵http://vrsverige.se/
6.5 Iterative Prototyping

In order to test the concepts from problems that were discovered in the interview phase, interfaces were designed and developed by utilizing iterative prototyping. Prototyping is a paradigm that is embodied in the “throw one away philosophy” of system development [51]. Iterative prototyping keeps refining a design by creating partially working prototypes and user-testing them at a fast pace to get early and regular observation of the system and interface behavior [52].

One of the main benefits of rapid prototyping is that the approach might reveal misunderstandings between designers and developers and the users, and are in line with the findings from empirical research about designers and how they work [53]. These misunderstandings can originate from differences in backgrounds and/or experience as Gomaa and Scott concludes in a study about using prototypes throughout a software development process. [54] The authors go on to state that creating a prototype in order to evaluate a product specification is both effective and less expensive than shipping straight to production.

This phase was divided into short sprints and carried out according to the design, test, evaluate principles from iterative prototyping. [52] The focus and software used for these steps can be found in Table 1. The created prototypes along with the results from testing are found at 7.2.5

6.5.1 Lo-fi

The first step when prototyping is often to sketch it up on a piece of paper. This is proven to be one of the best methods this early in the process as the rough esthetics provides a platform where it’s easy to create, easy to discard ideas and are more encouraging of criticism from test subjects. [55] After initial prototypes were made, the concepts were evaluated by drawing them in Tilt Brush. Doing this might remove some ideas just because what work on paper might not work in 3D and VR. There have been instances where VR games and projects have used the previous mentioned application Tilt Brush (see Section 5.6.2) for prototyping purposes. The company Dream On VR ⁶ is an example as used the application to prototype level and map designs ⁷ for their VR superhero game.

The testing of these prototypes for VR purposes comes with new challenges. For testing a design created for a screen, hand-drawn sketches of the different views and scenarios from the application are displayed one at a time to the user. When the user "interacts" with something on the sketch, a new sketch is displayed [55]. This is not possible on paper when testing for the purpose of VR.

The prototypes were created in large numbers of different concepts as sketches on paper. These concepts had different approaches according to the results of interviews (Section 7.2). All concepts were then evaluated based on the theoretical framework (Section 5) and through consultations with UX professionals working at North Kingdom. They were then tested in a ‘real-world’ scenario where the VE was substituted with the real world using an approach called ‘Wizard of Oz’, where a ‘wizard’ (test leader) simulates actions and behavior, of a system [56]. With this approach a system can be tested without full implementation. The test subjects were given a real

⁶https://dreamonvr.com
⁷https://uploadvr.com/til-brush-game-levels-prototype/
3 DOM controller and were instructed to perform certain tasks with objects in the real world. The UI of these tests was created on post-its (smaller UI components) and sheets of paper (larger UI). As the user’s interactions were not tracked, every move, point and interaction were explained orally as the test leader acted as the ‘wizard’ and manipulated objects and UI depending on user interactions.

### 6.5.2 Hi-Fi

When a basic understanding of the problem and the general concept had been developed, hi-fidelity prototypes were developed. The key difference in this step compared to the previous lo-fidelity prototype is covering more dimensions of the desired product. In their study about the validity of hi-fi prototypes, Virzi et. al concludes that this is a necessity in order to discover problems with a design if the lo-fi prototype lacks certain dimensions [57]. For this concept, the missing dimension is mainly the properties of a virtual reality that is lacking in the real world. Properties like teleportation, changing the scale of 3D objects, remote object placing etc. These prototypes were created as interfaces in Sketch\(^8\) and Adobe Illustrator\(^9\) which are software used for creating digital design elements and interfaces in vector format. All interfaces were exported as UI elements into Unity using the plugin Fetch\(^10\). This allowed for an interactive test-suite that was created for HTC Vive in Unity for user testing purposes. This was achieved using SteamVR\(^11\) and a template VR scene created by Ray Wenderlich\(^12\). The scope of these tests included the understanding of the basic functionality, proper testing of relative sizes and proportions, user flows and overall user experience.

### User testing and Evaluation

The prototype was evaluated through user tests and follow-up interviews. The purpose of this evaluation is to assess the usability and validity of this VR editor concept. The test was based on a series of tasks that the subjects carried out. Each task used at least one of the modules that were developed based on the initial interviews. The test subjects were developers, art directors and interaction-designers at North

\(^8\)https://www.sketchapp.com/
\(^9\)http://www.adobe.com/se/products/illustrator.html
\(^10\)http://fetchui.com/
\(^11\)https://support.steampowered.com/kb_article.php?ref=2001-UXCM-4439
\(^12\)https://raywenderlich.com/149239/htc-vive-tutorial-unity
Kingdom with different levels of experience of working with Unity\textsuperscript{13}. This approach allowed comparison between experienced and no experience users, as well as test if this concept could work as a mediator-tool in cross-functional teams. After each test, an interview was conducted in order to get qualitative data about the performance and how this concept works compared to the process that is used today. The prototype was limited to only handle the functionality for scenarios including the dimensions that were missing in previous tests. Some of these limitations could be handled by the test leader using the Wizard of Oz approach [56], by changing the VE from the computer that is running the test suite, as the test subject is immersed and unaware of actions that the test leader makes during runtime. The following tasks were included in the tests, and were chosen based on the basic functionality of Unity and from section 7.2:

- Move around in the VE
- Select an object
- Move objects to a certain position along their reference planes
- Move object along their third axis (move the reference plane). Stacking objects on top of each other
- Delete an object
- Create a new object that is not already in the scene
- Duplicate an object
- Add a property to an object

\textsuperscript{13}https://unity3d.com/unity
7 Results

In this section, the results will be presented. First the hardware evaluation of different HMD, then answers and insights from expert interviews. The part will be dedicated to the prototypes that were created based on the results from the previous sections.

7.1 Head-Mounted Display Evaluation

In order to get insights into possibilities and limitations of VR hardware today, two different HMDs were evaluated. The selection were based on availability of the HMDs and previous experience by the author. The focus of the evaluation lies in portability, DOF, possible interactions and support by Unity VR Editor. The selected devices are:

- HTC Vive
- Google Daydream

7.1.1 Hardware specification

Table 2 consist of the hardware specification of the selected devices.

<table>
<thead>
<tr>
<th></th>
<th>HTC Vive</th>
<th>Google Daydream</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resolution</td>
<td>1200 x 2160 pixels</td>
<td>(Google Pixel) 1080 x 1920 pixels</td>
</tr>
<tr>
<td>Nr of Controllers</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>Controller DOF</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>Room tracking</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Wireless</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Phonebased</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Unity EditorVR Support</td>
<td>Yes</td>
<td>No</td>
</tr>
</tbody>
</table>

Table 2: Hardware specification of HTC Vive and Google Daydream
Figure 7: Outline of buttons from HTC Vive controller.

**HTC Vive**

The HTC Vive is the most powerful and has the best resolution of the selected devices, which allows for more complex environments and experiences. It has two wireless 6-DOF controllers that consists of an application button (Label 7.a), a trigger (Label 7.b), a thumb touchpad (Label 7.c), two squeeze buttons (Label 7.d), and a menu button (Label 7.e). Its biggest asset, however, is the room tracking of both the HMD and the controllers with which the user can perform complex interactions with high precision. With support for Unity EditorVR and its API the Vive can be used to alter existing Unity projects. These features does, however, come with a cost, both in terms of portability and financially. The system relies on a standalone state of the art computer to run applications and experiences, which is connected with wires to the HMD. In order to track the user’s controllers and HMD, two sensors have to be placed in the room which creates a bounding box of the tracked area. These sensors have to be mounted on a wall or onto a stand and then calibrated, which in itself is a five-minute process. Its high price tag combined with space required limits the number of devices that will be present in a studio or office. This device is suited for high-precision interactions in complex scenes and environments, it also works as a great test suite for iterations of an experience as it could allow changes to the experience that is being tested.
7.1.3 Google Daydream

Daydream is created by Google as a phone-based VR device that works without external support. Only a selected number of mobile devices are currently officially supported as "Daydream-ready", and the performance of the setup relies on the specification of the mobile device. The device consists of a HMD that is tracked with internal sensors and one wireless 3-DOF controller. The controller offers a clickable thumb touchpad (Label 8.a), a application button (Label 8.b) and a system menu button (Label 8.c). The biggest advantage with this device out of the list of selections is the portability and space required to operate. This device can be used while at a regular desk at an office, at a price point that makes it individually available. This setup suffers from some shortcommings as well. It struggles when it comes to complex environments because of limited processing power in the mobile device. It also has a limited set of interaction capabilities due to the fact that the internal sensors in the HMD and the controllers cannot track relative motions in space, only changes in movement. It is not supported by Unity EditorVR which limits the functionality in practice to something like a sandbox, where the created environments stay within the enclosed system of the device or application. This setup works best for rough tests and basic interaction pattern, for simple prototyping and tests of elements and its basic properties. With this approach, the portable and simple setup can be utilized by professionals at their current workstation.
7.1.4 Unity VR Editor

The approach of using an HMD setup to alter a VR project is designed from an API that was released for the 3D game engine Unity in late December of 2016\(^3\). The tool is called Unity EditorVR and allows altering of an existing Unity project while immersed using an HTC Vive or Oculus Rift.

7.2 Findings from interviews

A lot of aspects that arose from the interviews are referenced in the Theoretical Framework (see Chapter 5), therefore these will not be showcased in this section. There were, however, answers and insights that did not appear before the interviews and were at most speculations. The results from the grounded theory analysis are described in this section.

7.2.1 The learning curve and programming knowledge

The majority of insights in this section are gathered from question 7: “Which are the biggest problems you meet in your daily work?”. One of these insights were that in order to work with a VR project as a designer/UX/developer or art director some level of programming knowledge is required. The reason is that in order to create or change something in a developed VE, it has to be done using a 3D game engine such as Unity\(^4\) and Unreal\(^5\). This has been mentioned previously in this study, but the ramifications and the differences of why this is an issue was not. From the interviews these differences can be divided into two parts, prototyping (read more in section 6.5)/concept creation and testing/fine tuning.

The first part consists of trying a concept or idea in VR on the fly, from something like a paper prototype. At the moment this requires a lot of knowledge of the software that the application is developed in, which is troublesome if you’re not a developer and without knowledge of this software. A UX designer that needs to test a UI interaction or animation can not effectively use the same tools that are used for screen designs. A tool for simple rough creation and manipulation requested in order to bring all types of designers up to speed in prototyping for VR. For a team in an early concept phase, it’s important to get the explore priorities and finding a balance between features and budget. By creating concepts early on, these estimates can be evaluated by creating rough prototypes in VR and analyzing complexity, resources any problems that can occur for each feature.

For the second part, the medium is slightly different. When the software is in the production phase, according to interviews with members of a VR core team, a lot of time is assigned to tweaking the existing VE and manipulating objects from the users’ perspective. This is troublesome when the adjustments are made on a computer, and the testing is done with an HMD. Interviewees explains how the constant switching between the two medium is both time-consuming and painful in the long-run, as the HMD creates chafings on their face and forehead from sliding the

---

\(^3\)https://blogs.unity3d.com/2016/12/15/editorvr-experimental-build-available-today/

\(^4\)https://unity3d.com/

\(^5\)https://www.unrealengine.com/what-is-unreal-engine-4
HMD on and off between tests. The optimal solution to this problem is explained as a tool that can be used with an HMD to tweak properties of the VE when immersed.

7.2.2 Time-consuming processes

When asked about the time and effort of each parts in their design process, there were two major issues across the board. **Visualizing and explaining the design** is really hard until it is created in VR. Furthermore the effect of this is a creative gap between the brainstorming (creating concepts) and the implemented design, it also limits the collaboration between team-members with and without programming experience. **Switching between mediums** when designing is inefficient as well as physically painful. As the current process is explained in two steps;

1. Creation and manipulation (on a desktop computer)
2. Testing and validation (with an HMD in VR)

Switching between mediums like this significantly extends the designing and development phase, and it keeps them out of their "zone". It also interrupts the user from creating a state of pure observing and testing. Another issue that arises is the significant amount of marginal errors as the validation results have to remain intact in memory during the medium transition and when correcting based on the validation. Is because of high cognitive load and correlates with the findings of Sweller that are explained in section 5.2.1.

7.2.3 Field-specific issues

When asked about flaws within the current design process and what features that a VR tool should consist of, there were similarities as well as diversity within the responses. By compiling all responses, the result display as a Venn diagram (Figure 9) consisting of the two parts that were explained in section 7.2.1. This provides an overview of what is expected for each part of the process. It can be argued that this segregation should push for a two-level solution for a VR tool.

7.2.4 Design Process and future

The design processes that were described have significant differences between parts of the industry, for the experience design teams it still consists of a lot of experimentation and concepts trials. According to the interviewees at North Kingdom, using a "Previz" which is a rough version of the experience, allowed the team to communicate their design ideas and feelings to a client. Several interviewees mentioned that one of the biggest uses for an immersive tool is when creating and modifying the previz. This would allow them to create and manipulate the previz in VR until they are satisfied, ship the previz and later build the final implementation with the previz as a foundation.

When asked about the future and what they look forward to seeing for VR development, the number one this across the board was the ability to make changes and test when immersed in the project that they are working on.
7.2.5 Features

None of the interviewees had tried a tool scene creation before, only creative applications like Tilt Brush (details in section 5.6). They did, however, mention features that they expect or require in order for them to work in VR. Ergonomically and practically, the consensus were that this tool should be usable from a sitting position. This would allow them to work from their current workspaces today and would be less exhausting over time. This correlates with previous research about human body limitations and VR, which is summarized in section 5.3. Below is a list of features that members of experience-design core teams would prefer to have a design tool for VR projects:

- **Overall**
  - Access objects from a library and create a scene
  - Manipulate objects
  - changing object parameters inside VR
  - Repeat animations while tweaking timings and distances

- **Prototyping**
  - Creating rough animations
  - Try out interactions and movements
  - Adding and adjusting sound and lighting

- **Development**
  - Place and change details (tweak) things in VE.
Apart from this, some parts were added as "not preferable". Most of these are part of the development phase and has a significant amount of scripts involved. According to the interviewees, these are better suitable for a desktop computer. From this, a Venn-diagram was created to categorize the features based on the phase and title of the user.

7.3 Vrify - A Virtual Toolbox For Experience Design

Based on the results from the interviews and the theoretical framework, a toolbox for creative processes and design of VR applications was created. The toolbox were designed based on simple user stories and scenarios. Specifications of the controllers and buttons that are referenced in this section can be found in section 7.1.

7.3.1 Lo-Fi Prototypes

Several different sketches and concepts were created to obtain a diversity of interfaces to test. A lo-fidelity prototype was developed with these sketches as a foundation, these were later evaluated with simple testing rounds. The concepts and elements in this section were selected for the Hi-fi prototyping phase after these tests.

Main interface - Belt UI

The main interface is designed for the user’s ergonomic position when seated, displayed from the hip of the user. (Figure 10) The main purpose of this interface is to create a new object and place them into the world (Figure 10b), as well as to change properties of selected object/objects (Figure 10a). The centered feature in this UI is the scene hierarchy, from where objects can be selected and created. The purpose of this approach is to introduce the hierarchy to the user as the starting point of the process. The users FOV will look like an extension of a workbench (Figure 13a), and can be accessed by pressing the application button on the controller.
User Interface Selectors

A selector UI (building from work in section 5.5.1) is used to manipulate and interact with objects. As seen in Figure 13b, the selector appears next to an object that has been selected and offers actions that are targeted to the selected object. The UI is designed as a pie menu (Figure 12a) that grants the opportunity one level of sub-menus (Figure 12b), similar to the interface of HoloSketch which is described in section 4. The selector UI is accessed through the touchpad on the controller (for hardware specs, see section 7.1), but is visible as a part of the VE, unlike the interface in HoloSketch which hides all other elements and objects.

Object selection and manipulation

Interactions and sequences for selecting and manipulating objects (Figure 11) are explained here as scenarios.

- **Selecting an object** The user points the controller towards object, so that the raycast hits it. Then the user clicks the selection-button to select the targeted object.

- **Moving an object** When the user selects the object, a bounding box becomes visible around the object along with three axis’ and a grid (Figure 10a). By
targeting the bounding box and holding the selection-button the user can now use the raycast to position the object along the grid. If the user wants to move the object in the third dimension (that is not covered by the grid), they grab the grid in a similar fashion and use the raycast to move. Its state is movable by default. Unless changed in the selector UI (section 7.3.1), it can be moved straight away. The axis’ of the grid can be changed using the selector UI.

- **Creating a new object** The user opens the Belt interface and selects the "New Component" button (in the hierarchy section), which displays a new view (Figure 11a). This view contains miniatures of all objects that are available to the user. The user finds the desired object, grabs it by holding the selection-button and points in the world to decide its position. the object is represented as a half-transparent version of itself, and is placed by releasing the selection-button.

- **Duplicating an object** The user selects the object that is desired for duplication then opens the selector UI, then selects the 'Duplicate' option. The menu closes and the user points the ray-cast toward the grid of the original object. An identical half-transparent object appears at the end of the ray, onto the grid. When the user is satisfied with the position, the selection-button is pressed to create the object in the selected location.

- **Properties of an object** The user opens the Belt UI, then selects an object with the raycast. The selection can be made on the actual object or on the object name in the hierarchy in the UI. When an object is selected, all properties for this object is displayed in the right section of the Belt UI (Figure 10a). A new property can be added by selecting the "Add property" button.

**Tilt-Brush Tests**

Concepts that made it through initial testing were drawn in Tilt Brush (see Figure 14) to visualise how perspectives and field of view work with the concepts in VR. Using the application, frames from different scenarios were sketched out, containing UI’s, objects and other visual elements.
7.3.2 Hi-Fi Prototype

Concepts and elements from the lo-fi prototypes were refined into hi-fi prototype. The hi-fi prototype is designed for mid-level and high-level VR devices, the interactive tests that followed were created for a HTC Vive. The prototype features elements from two main concepts/tools that were discovered.

The first is a tool for rough prototyping, as explained in section 7.3.2, that provides an intuitive and fast way to explore concepts without requiring previous scripting or VR development knowledge. This tool is to be used in the early stages of the experience design process (described in section 3.2). In later stages, when the concepts have been selected and are being developed, a tool acting as a test suite and with object manipulation is preferred. The hi-fi prototype along with these concepts are presented in this section, which also covers results from user testing and concept evaluation.

Main interface - Belt UI

The main interface is designed for the user’s ergonomic position when seated, displayed from the hip of the user. (Figure 16) The primary feature of this UI is to create new objects and place them in the world (Figure 15a) along with changing properties of objects (Figure 15b). Users interact with the UI by pointing the raycast at an element and pressing the selection-button. The centered feature in this UI is the scene hierarchy, from where objects can be selected and created. The users FOV will look like an extension of a work-bench (Figure 19), and can be accessed by pressing the application-button on the controller. The user can with direct manipulation (described in section 5.5.1) position the screens in the UI according to their needs, to create the personal workstation.

User Interface - Object Manipulation

A secondary UI (figure 18) is placed on the touchpad of the primary controller as seen in Figure 20a, its primary use is manipulating selected objects. The UI is designed as a pie menu (Figure 18a) that grants the opportunity one level of sub-menus (Figure 18b). The selection UI is hidden by default and can be opened by clicking on the touchpad on the controller (for hardware specs, see section 7.1.2, 7.1.3). The UI can be closed by pressing in the middle of the trackpad when it is open.
(a) An interface for adding new objects to the VE

(b) A list of properties that can be added to the selected object

**Figure 15:** Digital representations of object properties and selectable objects in the primary UI: Belt UI

**Figure 16:** Digital representation of the scene and properties screen in the primary UI of the tool: Belt UI
Figure 17: Digital representation of object manipulation UI

(a) Accessing a submenu of the UI

(b) Selection in the UI

Figure 18: Multilevel UI for object manipulation
User Input and Object manipulation

Interactions and sequences for selecting and manipulating objects (Figures 20, 21, 19) are explained in this section as user scenarios.

- **Transportation** The user holds the touchpad of the secondary controller, which then points a raycast from the controller. The user points to the desired position and releases the touchpad to teleport to the targeted location.

- **Selecting an object** The user points the controller towards the object, so that the raycast hits it. Then the user clicks the selection-button to select the targeted object.

- **Moving an object** When the user selects the object, a selection plane becomes visible alongside the object (Figure 20a). By targeting the object and holding the selection-button the user can now use the raycast to position the object along the selection plane. If the user wants to move the object in the third dimension (that is not covered by the grid), grab the grid in a similar fashion and points with the raycast to the plane that the object should be elevated/moved to. Its state is movable by default. Unless changed in the selector UI (section 7.3.2), it can be moved after creation or selection. The axis' of the grid can be changed using the selector UI. The user can use the direct selection button on the secondary controller to move an object using direct selection (explained in section 5.5.1).

- **Creating a new object** The user opens the Belt interface and selects the "New Component" button (in the hierarchy section), which displays a new view (Figure 15a). This view displays miniatures of objects that are available to the user in the scene (figure 21a). The user finds the desired object, grabs it by holding the selection-button and points in the world to decide its position. The object is placed by releasing the selection-button.

- **Duplicating an object** The user selects the object that is desired for duplication then opens the selector UI, then selects the 'Duplicate' option. The duplicated object is created at the current point of the raycast.

- **Properties of an object** The user opens the Belt UI, then selects an object with the raycast. The selection can be made on the 3D object with the raycast or on the object name in the hierarchy in the UI. When an object is selected, all properties for this object is displayed in the right section of the Belt UI (Figure 16). A new property can be added by selecting a property in the "Add property" button displayed in figure 21b.

Mid-level Concept

The interviews and the hardware analysis, as well as the theoretical framework in previous sections, proves the validity of a mid-level solution within certain parts of the experience design process. Limited to the functionality for rough prototyping and the technical constraints, a mid-level design solution is presented. This solution has the same basic functionality as the high-level but with some limitations and changes, which are explained in this section.
**Figure 19:** Field of view image of the main BeltUI interface

(a) Interacting with piemenu on controller  
(b) Moving an object with the raycast

**Figure 20:** Field of view image when manipulating objects

(a) Creating a new object  
(b) Adding a property to an object

**Figure 21:** Field of view image with different parts of the UI
• The VR application is not connected to an existing VR scene, and does therefore not support altering of scenes.

• Objects are selected with the touchpad.

• The pie menu UI that is described in section 7.3.2 is opened by holding down the touchpad.

• Selecting a "slice" in the pie menu UI does not trigger the action, but changes the mode accordingly, eg. When duplication is pressed when an object is selected the menu closes and the mode changes to duplicate. When the user points to a surface and clicks the touchpad a duplicate of the previously selected object.

• Moving an object requires the user to select the object (move-mode is selected as default upon selection), point to the desired position on a surface or reference plane and clicking the touchpad.

• Teleportation is located in the pie menu UI.

• There is no way to perform direct manipulation on objects.

• The main UI, Belt UI from section 7.3.2, Can no longer be repositioned. It will appear in the users general direction each time it is opened.

User tests and evaluation

Common errors, habits and insights from user tests of the hi-fi prototypes are listed in this section. There was no significant difference in adopting this concept between users with and without programming experience. This data is analyzed in section 8.

• The reference plane can hinder the user from selecting objects behind.

• Users cannot visually distinguish object positioning from reference plane positioning.

• Moving an object along the third axis (moving the reference plane) is not intuitive for users without experience or directions of usage.

• Placing object along the third axis by pointing at another surface causes confusion and is hard to grasp.

• Its hard to place object at the edge of a surface when moving the reference plane.

• The selected object needs to be highlighted in order to identify it.

• Objects can be hard to select in complex environments due to occlusion

• Using direct manipulation on objects is a valuable tool.

• Moving objects across greater distances is hard with only teleportation as transportation.
• Using teleportation as a tool for getting a closer look is interrupting the user in their creative process.

• Placing objects far away is hard if the reference plane is not (close to) orthogonal relative to the user.

• Having objects move upon selections causes unintended movements.

• It’s hard to place objects next to each other without "snap-to" functionality.

• No way of selecting multiple objects.

• The pie menu UI is practical and intuitive.

• The pie menu UI needs vibration for users to interact without looking at it.

• Duplication on touchpad click causes placement in unexpected places.

• No visual feedback upon creation, duplication or deletion of an object when the user is looking at the pie menu.

• Belt UI provides a comfortable workstation in a seated position.

• Interaction with Belt UI can cause awkward body positioning when using raycast.

• Text is hard to read, icons and graphics are easier to comprehend.

• Placing a new object causes unintentional placements.

• The concept is a really nice solution to the testing and manipulation problem when switching between HMD and computer screen.

• Users express the potential of this concept as a tool for fast prototyping in VR.
8 Discussion

This section consists of an overall discussion, an analysis of vital parts from the results and methodology that were used in this study is presented. A short analysis of the interview answers and the correlation between these are presented. The results from testing and evaluation of prototype fidelities are also discussed, along with possible changes and features based on behavior and ideas from participants during the testing sessions.

8.1 Interviews

After the interviews, it became clear that two different solutions should be implemented, in a best case. One of the biggest advantages of having two different tools is that a tool for a low-level medium (created for a less powerful device) this tool could be used by several people in parallel with the much more convenient and portable Google Daydream\(^1\). Unfortunately this was tweaked however as the VR editors that exists only support hi-fidelity HMD’s (HTC Vive\(^2\) and Oculus Rift\(^3\)) and would therefore require a different implementation approach.

8.2 Lo-fi Prototype

Creating a lo-fi prototype for VR interfaces and VR concepts required another approach then for regular screen applications. A significant part of this phase was dedicated to finding an appropriate way of creating, validating and testing. Using TiltBrush to visualize a hand-sketched concept was a great technique for getting relative sizing and perspective to scenarios containing these concepts. The use of real-world objects and a real HMD controller as part of the test suite worked extremely well when evaluating concept sketches with a minimal overhead. By using post-its on objects, their current state was displayed in a fast and easy way.

The initial user tests of the lo-fi concepts were perhaps the most game-changing in terms of major changes to the prototype. Since the tests consisted of the users trying to interact with the tool and the environment, they are not informed on what kinds of interactions that are available. This gives a better view om the logic of what the user expects from the environment. For example, The initial concepts that involved movements on the trackpad of the controller for selection within a pie-menu(figure 1). With knowledge of the interaction patterns, the concept seemed really thought-out and logical, as well as it included ease of use. When testing

---

\(^1\)https://vr.google.com/daydream/
\(^2\)https://www.vive.com/eu/
\(^3\)https://www.oculus.com/rift/
these features however, none of the users actually tried to use this technique without hints from the test leader. Instead, they tried to point to items on the menu with the ray-cast (Section 5.5.1) until they were told that nothing happens. This result strengthened the idea of pointing instead of using the touchpad, and when users were asked why the came to that conclusion, the answer were because they pointed and clicked on the menu to open it. At the time of testing there were no cues to actually introduce the possible interaction with the touchpad and because of that, this approach was still carried into the hi-fi prototype as a concept as well.

Another insight unexpectedly appeared during the user testing of the lo-fi prototypes was the importance of the scene hierarchy. In this, all objects of the scene are listed as subobjects of each other. Initially, the design did not include this, as to eliminate all complex features that could distract the user from their primary objective, to create. Without this however, the sequence of creating a new object gave the user no relation between the new object and the rest of the environment. The usage of the hierarchy also enables users to interact with and manipulate occluded objects.

8.3 Hi-fi Prototype

Creating the hi-fi prototype was in itself a demonstration of the need for tools like the one explained in this study. The transitions between Illustrator, Sketch and Unity was pretty smooth but time-consuming. The majority of hours spent on this testing suite were creating the scripts and logic for all features and functionality of the concept. Using Wizard of Oz to minimize the amount of time on feature implementation was a great technique for this medium and methodology, as the user when immersed is unaware of the wizards’ actions. The advantages of this was however not recognized until the pre-test run. In retrospect, more features in the test suite could have been designed to handle actions from the test leader (wizard) instead of a full implementation.

The results from the user test of the hi-fi prototype surface a number of issues and behaviors based on the current design and concept, some of them connected to the dimensions that were missing in the lo-fi testing. This displays the validity of hi-fi testing for VR concept designs as it recovers issues that would otherwise remain hidden, and correlates with the reasoning in section 6.5.2. After conducting tests on the lo-fi prototype, the selector interface was prioritized higher than a controller-based interface for access to object manipulation tools. This was due to its direct connection with the object and that it was easier to understand when testing in the real world. When the hi-fi prototype was developed and initially tested, this changed again). When the user selects an object that is far away, the selector next to it appeared so small that it became very hard to read and interact with it. Perceptive scaling were added to the interface to avoid this, which caused the interface to occlude other objects adjacent to the selected object (more about occlusion in section 5.3.2). These problems shed some new light on the previously ignored solution, a controller-based interface using the touchpad. The interactions with this interface were chosen because they can be achieved using one controller, which supports both middle- and high-level devices (more information about devices in section 5.1), and its accessibility. The apparent issue with this approach that was discovered in lo-fi
tests was addressed by changing the color of the selected object when the menu is activated. The idea is to connect the touchpad interactions with a specific color, that separates it from the primary interactions done with the raycast.

The controller-based UI (pie menu) was a good solution for object manipulation, as users adopted the interaction style quickly and commented on the practicality of the concept. However, there were issues concerning the interaction patterns with this UI. As the action of each menu item is triggered upon selecting, the subjects had a hard time focusing on both the selected object and the menu itself in order to get feedback on the action. One solution for this is a mode-based interface; Where an action triggered with the selection button and the triggered action is based on the current selection in the pie menu. This approach would also solve the practical complications with object selection that appeared, even though the selection technique worked well in general. The issues involve a user selecting an object and it gets automatically grabbed, which lets the raycast ignore the surface of the object. In practice the object will instantly move to the surface behind it, as that is what the raycast hits. By using a mode-based UI from the pie menu, the object would first have to be selected, the grabbed in order to move it. Another way of solving the focus issue is by displaying a mirrored graphic of the pie menu on the object itself (similar to the selector that was presented in 7.3.1 but without raycast interaction)

In the prototype that was being tested, the objects were not highlighted upon selection. The only indication was the appearance of the reference plane. Adding a bounding box to the selected object when selected or moved is preferred. A similar issue appears when the reference plane is grabbed with intention of moving it, as there is no visual feedback that the plane is grabbed. This caused the users to re-grab the plane multiple times in confusion. The confusion was increased when the plane did not move based on rotational changes of the controller. Several times during the testing sessions there were scenarios where the user would try to elevate an object from the floor (with a horizontal reference plane) by increasing the rotation of the controller slightly, which still leaves the raycast hit on the floor. Because the reference plane will position itself according to the surface that the raycast hits and not the difference from the original rotation of the controller, the object will not move. This was mainly the plane had no desired surface as a reference and an issue that the majority of participants encountered, when discussed some enhancements to this element were discovered:

- Introducing two **supporting planes** on the remaining axis allows the user to manually position the plane by tracking the hitpoint on these planes. This feature would come with an option to toggle the visibility of supporting planes, depending on needs of the user.

- Tracking change in **spatial position of controller**. Several instances of this behavior were found when a test-subject tried to change the elevation of a box on the ground by lifting on of the controllers higher into the air. This proves to be the go-to interaction for first-time users but does come with practical issues, one of those being that the movement is limited to the physical reach of the user.

- Introducing an **axis arrow**. As test subject initially thought that changes
in rotation of the controller are relative to the position of the reference plane, introducing that as a possibility could add to the practicality of moving a plane to a "floating" position.

Trying to place objects next to each other was also a recurring issue during the test sessions, as the test subjects wanted to place an object pixel-to-pixel with another object using the raycast. This was difficult at longer distances as the raycast hit would oscillate over the targeted point, which could send the object flying between different surfaces. What was requested some users is adding snap-to functionality when pointing close to an edge, that would allow edges to work as magnets, sticking to each other when they get close to each other.

8.4 General Discussion

The tools presented in this thesis have potential to make significant improvements to the experience design process for VR project. By continuous iteration of the current prototype along with changes suggested in this section, this tool could see commercial using in a variety of fields. The simplicity in the design makes it available to more than just programmers and designers, as the evaluation showed no significant difference adoption of this tool between subjects with and without programming experience, it proves to be a plausible tool for communicating ideas and concepts in cross-functional teams. In its current form, some of the features actually performed better on participants with less experience in 3D developing tools like Unity. The cause of this might be that the expectations of a user with no previous experience are narrowed down to functionality in softwares that are created for a computer. The appearance of expectation through previous usage is a double edged VR sword as it can be utilized to give professionals functionality that they know how to interact with. On the other hand, doing this could mean that their expectations are based on the functionality that they are used to which can create uncertainty and confusion.

The hardships during the design and development of these tools are in itself a verification of the problem at hand, and the current need for a tool like the one presented. It is, however, hard to know where the industry will be in the foreseeable future and which role VR will have in that future. With evolved software and hardware could improve the precision and portability of the VR industry that could change the field of VR as we know it today.

The response from being in a seated position while working in VR was very positive. Several test subjects explained that they never really thought about the fact that they were seated or that it limited them in any way. They also utilized this to improve the accuracy of the raycast selection, by resting arms or controller on their legs or armrest. Using teleportation as a transportation technique also works well from a seated position but could be complemented with zoom or scaling of the world. Scaling the world around the user could benefit the process of moving objects in larger environments and handling larger objects. A zoom feature would keep the user in the same position and only increase the scale of their FOW, which could help users to make small changes on small or far away objects. The main UI (BeltUI) was also commented as a good fit for the seated position, along with some remarks about improvements to the interaction technique. Some test subjects found the raycast
technique to be awkward, forcing them into less ergonomic body positions. Based on this, some improvements that surfaced were to use the trackpad to navigate between objects or have the controllers act as drumsticks and hitting the UI buttons like drums. As a part of the ergonomic perspective, the size, weight and shape of current HMD can be straining on users neck and skin. If the increased interest in VR keeps going, further improvements to this part can be expected.

There are a lot of features that can be added to a concept like the one presented, and it is mostly a matter of adapting the concepts in a way that suits the medium. The biggest challenge today is to not getting stuck in old tracks and reusing features from a computer (2D) environment. Apart from the features that are present or were presented in this section, there were a few that were discussed during tests and could be good additions to the current concept;

- Animating an object being deleted och duplicated through the raycast ray. This would act as additional visual feedback for the user.
- Drawing an outline around several objects with the raycast for multiple selections. Objects that are inside the cone that is created between the controller and the marked area will be selected. A reference plane will be joined to the selected area in order to re-size depth.
9 Conclusion

This study highlights important issues in the current process of designing experiences for virtual reality. It recognizes the gap between the design medium (desktop computer or laptop) and the target virtual environment (using an HMD), and the friction that professionals within this field of work are experiences. Having to switch between these two mediums is both time-consuming and causes an unnecessary amount of errors, which correlates with the effects from increased amount of cognitive load. Furthermore, a design tool for VE (Virtual Environments) like the one explained in this study prevents the computer-to-VR transitions that otherwise exists, which can prevent ‘Flow’.

Based on the experience design process and the VR technology available today, this study has distinguished that utilizing VR is valid in certain phases of the process. Two fundamental tools have been identified, designed and tested based on these findings; In the early stages of the experience design process, a tool for rough prototyping provides an intuitive and fast way to explore concepts without requiring previous scripting or VR development knowledge. When the concepts have been selected and are being developed, a tool for object and scene manipulation of the actual concept in real-time (using Unity EditorVR or otherwise) is preferable. These tools are proven useful for professionals with and without previous programming or design knowledge and can therefore work as a medium for explaining rough ideas among cross-functional teams.

Furthermore, the usage of rapid prototyping and user testing with a Wizard of Oz approach throughout this study has proven to be beneficial as it the time and resources spent of each iteration, without compromising the validity or user experience.
Virtual reality is a rapidly developing today and has been doing so for the past 5 years. What we see today is only a glimpse of what will become of this field in the future. As VR evolves as a field, the ways of designing for VR will have to evolve with it and not get stuck in past ways. Utilizing the technology in ways like this study is crucial for professionals working with VR in the future. Experience design will be an important part of the field of VR, and will need to be accounted for in future attempts to redefine the process of creating VR content. The concept presented in this study, VRify, proves the possibility to solve some of the challenges that the industry is faced with today. In order to become a viable product in this evolving industry, it will need further improvements and will need to utilize an API like Unity EditorVR.
Bibliography


A VR Professionals - Interview Questions

A.1 General Info

Interviewee name:
Date:

A.2 Dictionary

Word: Explanation

A.3 Warm Up / Introduction

Where do you work?
What’s your title?
How long have you been working there?

A.4 Body

How many VR projects have you been involved in?
What are the problems you meet in your daily work?
What kind of design process do you append to VR projects?
Are there flaws with this design process in VR projects?
Are there any time consuming processes in the pipeline?
How would you like to work in the future?
How does the communication within the project core team work?
Especially between VR devs and designers/rest of team.
How are prototypes utilized in your VR projects?
How do you describe a VR concept or world to your teammates? (Which tools do you use?)
Have you tried applications where you manipulate the VR world?
What do you think about developing/prototyping for VR while emerged?
What do you think of this kind of tool within your current pipeline?

A.5 Conclusion

That was all the questions I had, do you have any other thoughts?
Thank you so much!
B Participants

- **Oscar Eriksson** Interview over Google Hangouts between the Stockholm and Skellefteå office at the 22nd of February.
- **Elin Sjöström** Interview at byBricks office in Stockholm. Conducted March 2nd.
- **David Ljunghill** Interview at North Kingdoms office in Stockholm. Conducted March 14th.
- **Joel Ring** Interview at VR Sveriges office in Stockholm. Conducted March 13th.