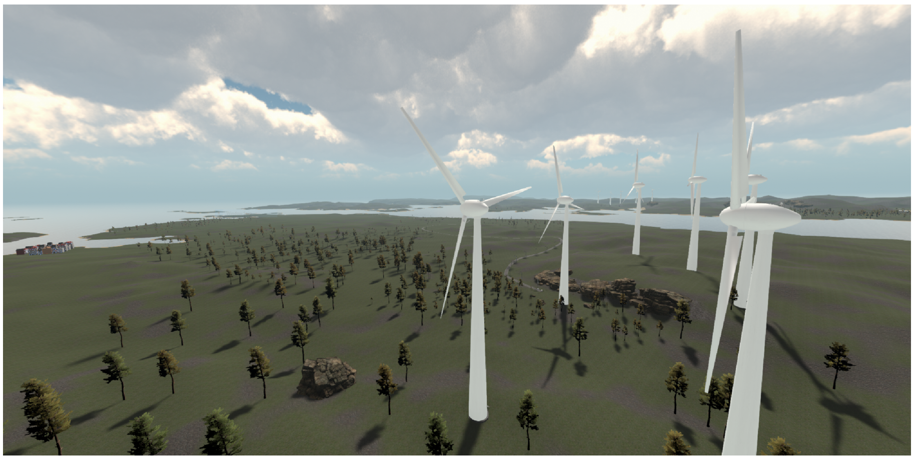
Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Computer Science

Jørgen Henrichsen

Engaging Young Job Seekers with an Internship as a Wind Turbine Technician in Virtual Reality

Master's thesis in Computer Science
Supervisor: Monica Divitini
June 2019



NTNU
Norwegian University of
Science and Technology

Jørgen Henrichsen

Engaging Young Job Seekers with an Internship as a Wind Turbine Technician in Virtual Reality

Master's thesis in Computer Science
Supervisor: Monica Divitini
June 2019

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Computer Science

NTNU
Norwegian University of
Science and Technology

Abstract

It is essential to include everyone in our society so that as many as possible can contribute to the welfare state. Youth are increasingly digital, and both companies and governmental institutions should communicate in a way that the youth understand, and that engages them. Utilizing virtual reality combined with game technology, immersive experiences that reach new generations of workers can be created, deemed as *virtual internships*.

In this thesis, an existing virtual internship is studied, and it is found that the users see substantial value in such an application, but that there is room for further development and experimentation to fulfill the youth's needs and expectations. A workplace, windmill technician, is modeled and implemented with further focus on some areas; guiding, tutorials, feedback, and graphics. This is found to make the application easier to learn and use, but not increase the perceived utility. Also, it is experimented with developing such an application on a different foundation than earlier, which is found to enable rapid development. The process when implementing a virtual internship is formalized and described in detail, resulting in a five-step process from reality to virtuality.

Sammendrag

Det er viktig å inkludere alle i samfundet, slik at så mange som mulig kan bidra til velferdsstaten. Unge mennesker lever i en digital verden, og både næringslivet og statlige organisasjoner må kommnunisere gjennom et medium de forstår, og som engasjerer. Ved å kombinere virtuell virkelighet med spillteknologi kan man konstruere oppslukende opplevelser for å nå nye generasjoner med arbeidstakere: dette refereres til som *virutal internships*.

I denne oppgavene studeres et allerede eksisterende virutal internship. Resultater viser at brukere ser en betydelig verdi, men det er rom for forbedring på flere områder for å oppfylle ungdommens krav og forventninger. Et nytt yrke, vindmølletekniker, er modellert og implementert med fokus på noen områder: guiding, opplæring, tilbakemelding og grafikk. Resultater viser her at applikasjonen er lettere å lære og bruke, men ingenting tilsier at det øker den oppfattede nytteverdien. I tillegg utvikles applikasjonen med bruk av et annet programvarebibliotekfor enn tidligere, som viser seg å tillrettelegge for hurtig utvikling. Utviklingsprosessen blir formelt redegjort for og beskrevet, og resulterer i en femstegs prosess fra virkelighet til den virtuelle verden.

Preface

This thesis is the final project for a 5-year Master Degree in Computer Science at the Norwegian University of Science and Technology. The thesis will look closer at using virtual reality technology to help and motivate young people in their career choice.

This thesis would not be possible without help. First a big thanks to Ekaterina Prasolova-Førland and Mikhail Fominykh for help and guidance in the development, testing, and writing. Additionally, a thank you to Monica Divitini. People at NAV have been extremely helpful with a special thanks to Heidi Fossen for continued support and assistance throughout the thesis. I am also grateful to all the different users from NAV that have tested the applications.

At TrønderEnergi a special thanks go to Charlotte Wulff. Furthermore, thanks to the rest of the helpful employees at TrønderEnergi for providing information about tasks and data from 3D scans of the turbine. Additionally, gratitude to the wind turbine technicians at Bessakerfjellet Wind Park.

Trondheim 07.06.2019
Jørgen Henrichsen

Contents

List of Figures

List of Tables

Abbreviations

SDK	=	Software Development Kit
HMD	=	Head Mounted Display
AR	=	Augmented Reality
XR	=	Extended Reality
VR	=	Virtual Reality
FPS	=	Frames Per Second
NAV	=	Norwegian Welfare Administration
MVP	=	Minimum Viable Product
FOV	=	Field of View
OEM	=	Original Equipment Manufacturer

Chapter 1

Introduction

New technology enables us to work in new ways, utilizing innovations to increase quality and become more efficient. In the coming decades, Norway will have a growing elderly population and an increasing life expectancy (SSB 2009), which means we need as many people as possible to participate in our society, as efficiently as possible. We develop and use new technology to make both our working lives and daily lives more comfortable and more efficient, but maybe technology can enable and motivate weaker groups of our society to participate in a larger scale? We can use technology to induce motivation, inspiration, and proper guidance for young peoples career choices. Today's youth are digital natives, surrounded by games and social media from the very day they are born, and it is adamant that the society reach out to this generation in a way that engages them. The number of people that play games has increased drastically, and the youth of today grow up with games as a natural part of their lives. A study from 2015, ESA (Entertainment Software Association) found that 155 million Americans play video games, with 51% of households owning a dedicated gaming console and 4 out of 5 households owning devices used to play games (ESA 2015). Virtual reality is one of the latest entries to mainstream consumer gaming technology, and does not only provide us with new immersive entertainment experiences but can bring new ways to learn, work, understand, analyze and communicate.

1.1 Background

This thesis is a part of the *Virtual Internship* project, which is a collaboration between Innovative Immersive Technologies for Learning (IMTEL) VR Lab at the Norwegian University of Science and Technology (NTNU) and The Norwegian Labour and Welfare Administration (NAV). The project explores new ways of reaching out and communicating with young job seekers through digital experiences. Games and technology are a big part of young peoples lives, and using technology to communicate effectively can be cost-effective and motivating for the users. The goal of the project is to explore the possibilities for utilizing virtual- and augmented reality for career advice (Øygardslia et al. 2018). The applications developed in this project aim to prepare, guide, and help young job seekers into the working life by letting them experience several workplaces virtually.

Virtual internships are not to be confused with regular workplace training applications, as the goal is not to *learn how to work or complete a task*, but rather *learn about the workplace, tasks and the experience of the job.* Although these to overlap, they aim for different learning outcomes as a result of the usage. By utilizing new technology and game elements, virtual internships aim to motivate and inform, especially young people, to explore multiple career choices and feel more prepared and ready to face the employment market. By doing this in a virtual environment, a safe setting for exploration and training can be produced. Additionally, this can be used as a platform for recruitment by companies, aiding them in displaying a more real story of how it is to work at the company. Details of four applications in the project, which is either developed or under development, can be seen Table 1.1. It is important to note here that the Interview app is not deemed as a virtual internship, but rather a training application for the job interview.

Application	Description	Status
FiskeVR	The user gets to experience working at a breeding facility and processing plant for salmon (Prasolova-Førland, Fominykh, and Ekelund 2019).	Developed
Interview	The user gets to practice the job interview situation (Fominykh and Prasolova-Førland 2019).	Developed
Construction Work	The user gets to experience construction work	Under development, bachelor project.
Windturbine VR	The user gets to experience the work as a wind turbine technician	Under development (this thesis).

Table 1.1: Overview of the current applications developed and under development for the project.

Øygardslia et al. (2018) evaluates the project in a report, and finds that the general feedback of the applications is positive, and job seekers can see the appeal and possibilities for using technology in this manner (Øygardslia et al. 2018). The report presents some general feedback and recommendations for further development and improvement of the project and the virtual internship applications. In Table 1.2 these are presented and linked to each other where applicable. From the feedback and recommendations, the potential for these virtual internship applications can be identified, but there is still room for improvement.

Table 1.2: Overview of general feedback and recommendations given in the evaluation report (Øygardslia et al. 2018)

Feedback	**Recommendation**
Both NAV-employees and NAV-users are postive to the concept of virtual internships.	Develop the applications using advanced VR/AR to reach the potential the users see in such an application.
The main appeal of virtual internships is the possibility to get to know a workplace, attaining information and creating a sense of achievement in users.	Further evaluate if such applications can increase the users sense of achievement.
Uncertain what game elements should be included, but cooperation and immersive stories seems like exciting opportunities.	Test implementing more game elements with most potential to evaluate their effects.
It is important that the workplaces and situations seem realistic.	–
Uncertainty whether 360-videos or animated graphics are preferable.	–
Participants think that AR could have a huge potential for virtual internships as well.	Further explore possibilities that AR technology can have in virtual internships.
–	Develop a platform for exchanging experience between NAV-employees.

Some of the other main takeaways from the report is that the users have tendencies to a general sense of achievement and safety when it comes to the process of applying for jobs. An essential factor for the users is that the applications can give them useful information and an experience of being at the workplace, which in turn gives them an ability to display their knowledge about that workplace for a possible interviewer. Also, users have a stronger feeling of safety when it comes to the choice of career if they have the chance to try it out in a virtual internship before applying for a job. The report sums this up in two main points: *the desire to get the job one applies for* and *the desire to not be erroneously employed*.

A more recent evaluation of the project by Prasolova-Førland, Fominykh, and Ekelund (2019) states that the general feedback for these applications is positive, both from NAV users (job seekers) and NAV employees (welfare professionals) (Prasolova-Førland, Fominykh, and Ekelund 2019). Important aspects for consideration are mentioned, which is summed up in Table 1.3. Notably, further research on the methodology when developing these applications is found to be needed, in addition to more workplaces represented. The need for more investigation into the development process is the basis of this thesis, which will be developing a new virtual internship where insight into the development process will be central. Additionally, during development, more guiding will be utilized in the application to see if this has any effect during usage of it. The report also found that it is vital that the applications are correct from the industry point of view, which means a connection to the industry during development is necessary. Before developing a new virtual internship, and the existing application, FiskeVR, will be studied to gain deeper knowledge in the field. The two studies can be read more about in Chapter 3.

Table 1.3: The main findings in an evaluation of the concept (Prasolova-Førland, Fominykh, and Ekelund 2019).

Topic	**Result**
Gaming elements	Should not be too entertaining, while still utilizing elements from games to stay interesting.
Feedback from environment	Need more feedback from the environment on what actions is right and wrong.

Continuation of Table 1.3	
Topic	**Result**
More workplaces	Need more suitable workplaces to explore the full potential.
Correct information	The workplaces need to be correct from the industry point of view.
Methodology	Need to furhter develop a methodology for developing future job taste applications.
Video vs 3D	360 videos are much cheaper than digitally modelled experiences, but they lack in interactivity that is appreciated by the users. A combination could be the solution.
Training of NAV employees	NAV employees (welfare professionals) and teachers must be trained in usage of the equipment needed for the applications.

1.2 The wind power industry

The developed virtual internship in this thesis is for wind turbine technician. This career is relevant, both locally in Trøndelag, in Norway in general, and it is a workplace for the future. Just in central Norway, six windmill farms will be entirely constructed by 2020, with three owning companies (Statkraft 2018). The project is the largest on-shore wind farm project in Europe. The potential total size of the project is 1057 MW in total, for all six farms combined with a total of 277 wind turbines. In 2017 wind power was the power source that increased the most in Norway (SSB 2018). In the world as a whole, wind power is currently supplying 5% of the global power demand (WWEA 2018). The wind power industry is therefore deemed an interesting opportunity, and it is a workplace that can be hard to understand without trying it. Additionally, it can be seldom or not possible to get access to a wind turbine in order to see it. Presenting the workplace virtually can therefore make it more available for young people.

1.3 Research Questions

The purpose of this thesis is to further gain knowledge about the development process, design choices, and usefulness of a virtual internship. The design and development of a virtual internship are driven by data collection and transformation from a real-life workplace into a virtual representation of that workplace. Tools and steps are both presented and assessed. Virtual Reality Toolkit, a platform-agnostic SDK for development of VR applications, that is not used in the project earlier, will be used and evaluated in order to explore its usefulness. An increased focus on in-game-guides that tell the player what to do and how to do them, tutorials and direct feedback will be experimented with, and to what degree this impacts the perceived usefulness and user-friendliness of the application. The research questions are as following:

- **RQ1: What is the current value of virtual internships?**
- **RQ2: What tools and processes are useful and necessary for creating a virtual internship and representing a workplace realistically?**
- **RQ3: How do more focus on information, in-game guiding, and direct feedback impact the perceived usefulness and user-friendliness of a virtual internship?**
- **RQ4: What advantages and disadvantages does Virtual Reality Toolkit bring to the development of a virtual internship?**

1.4 Contribution

The contribution of this thesis to the project will both be further insight into how the concept of virtual internships are received by users and more specific insight into the development process of a virtual internship. A new, locally relevant workplace has been simulated using information and data from TrønderEnergi to properly recreate the workplace, along with validating the realism with professionals (wind turbine technicians). Specifically, this thesis contributes to the virtual internship project as a whole as follows:

- **Developing a new virtual internship**: This application can be further used to research the effects of the actual usage of such applications.
- **Definition of the process**: Detailed description and definition of the process of recreating a real-life workplace in virtual space. This results in a 5-step digitization process, from reality to virtuality.
- **New foundations**: The advantages and disadvantages of developing such a virtual internship with a framework that is not earlier used in the project are investigated and presented (specifically Virtual Reality Toolkit).
- **Guiding**: Apply more in-game guiding on interactive elements to see if this makes the app easier for the users, and if it makes them perceived as more useful.
- **A new menu system**: A menu that is spawned in front of the player in world space, with a list of tasks, a map, list of gear and overview of the current score. The menu system is flexible and possible to reuse for future applications.
- **More direct feedback on completion**: Give users direct feedback in the form of pop-up notifications when a task is completed.
- **Dedicated tutorial**: A dedicated tutorial that teaches the user how to teleport in the application. This can be further expanded on by future projects to include more concepts.

Chapter 2

Literature

This chapter will present relevant literature and theory, related to both virtual reality technology and usability concepts, and game design theory. Related work will be investigated to identify earlier experiences and results in using virtual reality for instructional and teaching purposes, specifically career guidance and workplace-related informational applications.

2.1 Virtual Reality

Virtual reality has existed as a field of research for decades (Jerald 2016), but still has not truly broken into the mainstream market (Fogden 2018). It has, however, been gaining more traction (Thomson 2019). Before studying technology and techniques to create and present virtual environments, it is useful to understand what virtual reality is.

Some scholars define virtual reality only by what the technology is doing to describe what *it* is. Virtual reality is, then to some extent, merely a computer-generated environment which can be interacted with, and produce feedback to the user through multiple modalities (Burdea and Coiffet 2003), (Jerald 2016). Now, this direct definition of the concept is quite tangible but is not enough to understand the real possibilities that virtual reality possesses. What kind of experience is it conveying? Moreover, should the definition be limited to what today's technology can produce?

Steuer (1993) emphasizes the importance of not just defining virtual reality through technology and hardware, though we will look more at both technology (Section 2.3) and hardware (Section 5.3) later, but through the experience it can provide. Steuer criticizes classifications of virtual reality that includes the technology in use when defining the concept. The key to virtual reality is *presence* (Steuer 1993), or *being there* (Reeves 1991). If the real, physical world is mediated only through the human senses, it is deemed a *first order* mediated experience. *Presence* then occurs if a *second order* mediated experience is experienced, partly or wholly, as a first order mediated experience. That is when, an experience not only mediated by the human senses, but also by technology, is experienced as if it were only, or partly, the human senses mediating the experience to the individual. It is important that presence is not a property of the technology, but a property of the individual that the technology can evoke (International Society for Presence Research 2000).

These mediated experiences are described by Richey (2018) as an experience that is experienced second-hand (Richey 2018), or second order as ISPR puts it (International Society for Presence Research 2000). Steuer, on the other hand, does not define mediated experiences as second order, but simply as mediated since the real world is not mediated by any technology in the first place (Steuer 1993). The definitions and terms of these concepts do vary, but the essence is the same: an experience, mediated through technology, is wholly or partly, experienced as if the technology was not involved. It is imperative to understand that it can and will most likely be a *partly* occurrence as complete presence naturally is less likely to occur, especially with the technology we have today. The point is virtual reality is a system, presenting an interactive and vivid virtual world, where presence is the key concept that it can induce in the users.

2.1.1 The degrees of virtuality

Not all experiences are completely virtual, and there are degrees of how much of the experience is made up of virtual entities, or content. It is useful to discuss the different ways reality is mediated by extended reality(XR)-systems to form a basis for understanding. XR is a collective term for all environments consisting of virtual elements, either mixed or not with real-world elements. XR is a term not much used in literature, but popular in the industry and commercially as a collection term for these experiences (Qualcomm 2017; Accenture 2018; Jehan 2018). Milgram (1995) describes how the notion of a varying degree of virtuality can be laid out on a reality-virtuality continuum (Milgram et al. 1995). Figure 2.1 illustrates this continuum with the real environment and virtual reality (VR) at each of the extrema.

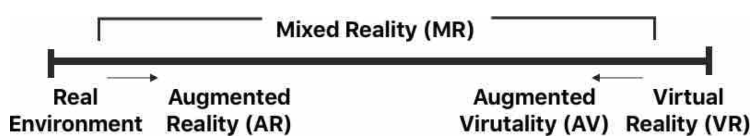


Figure 2.1: The reality-virtually continuum described by Milgram et al. 1995.

The real environment is an environment consisting solely of real objects, while virtual reality (VR) is an environment consisting solely of virtual objects. In between the extrema, there is mixed reality (MR), where both real and virtual objects coexist. The two groups of systems under MR is augmented reality (AR) and augmented virtuality (AV). Milgram defines AR as: “augmenting natural feedback to the operator with simulated cues” (Milgram et al. 1995). The reality that the user of such a system sees is manipulated or added extra content, and the visual and auditory feedback can differ from the actual real-world environment of the location of the user. Figure 2.2 displays an example of AR, with a digital chair added to the real world environment of the user.

Milgram continues to define AV as “completely graphic display environments, either completely immersive, partially immersive, or otherwise, to which some amount of (video or texture mapped) ’reality’ has been added.” (Milgram et al.

Figure 2.2: An example of a mobile application using AR to insert a digital chair in the room (John 2018).

.

1995). The problem with separating AV and VR is the fact that almost any virtual environment will have some amount of reality added to them, through textures[1] or video material. Therefore, from this point on, AV and VR will be termed simply VR to lessen the confusion. From this discussion, the terms to be concerned with will be AR and VR, with XR as a general collection term for all these experiences.

2.1.2 Interactive, Immersive and Imaginative Experiences

It is important to understand what virtual reality enables, and what makes an experience or application seem engaging and worthwhile to use. Burdea and Coiffet (2003) list three key elements of a virtual reality application: *interactivity*, *immersion*, and *imagination* (Burdea and Coiffet 2003). Kirsh (1997) defines interactivity as reciprocal actions, but not necessarily between two agents with the ability to act. It can just as well be between an agent,

[1]2D graphics applied to 3D environments. Can be drawn images, or photographs of real-life objects like wood, concrete, or other materials.

and its environment of action determining what can be done and what happens as a result of the agent's actions (Kirsh 1997). It is a property of the world presented, where the environment responds, changes, or impacts future events in the application (Jerald 2016). The user can alter objects or events and receive feedback when interacting with the environment. When it comes to immersion, Sweetser and Wyeth (2005), in their game-literature-focused interpretation of the traditional flow-theory (Csikszentmihalyi 2014) maps immersion to "Deep but effortless involvement, reduced concern for self and sense of time" (Sweetser and Wyeth 2005). Immersion is a phenomenon where the user becomes less self-aware, feel emotionally and viscerally involved and even experience an altered sense of time. Important aspects for immersing the user are good quality and vividness in the graphics and audio, and the extent to which this "surround" the user, e.g. Field of View. Additionally, in VR, the congruence between what the user feels, and what they see is crucial. When the user moves their head, the display should update accordingly, so that it looks like they move their head in the application as they do in real-life (Jerald 2016). Imagination is more a property of the user than of the technology or the application. The extent to which a virtual reality application is able to perform well, also depends on the human *imagination* (Burdea and Coiffet 2003). The technology only takes the experience part of the way, and in the end, the user and the user's context, state of mind and perceptions also affect the ultimate experience the user has of that application. In some regard, one can say that imagination, as described here, is the key to induce presence from immersion. Human senses and perceptual processes mediate the experience of the world (International Society for Presence Research 2000), therefore the occurrence of presence in an environment will depend on the user's senses and perceptions. Immersion can enable presence, but it depends on the user's context and perceptions (imagination) whether they feel *presence*. Presence is a function of immersion and the user; immersion is capable of producing presence, but it does not mean it will happen for every user (Jerald 2016).

The concept of presence is an important contribution by virtual reality to career guidance and workplace simulations. Villani et. al (2012) in a study of 20 people, found that it is possible that the experienced *presence* is higher in virtual reality than in real-world simulations when comparing virtual reality job interviews to real-world simulated job interviews. The experience of presence (*being there)*, might be influenced by the feeling of *making sense there* (Villani et al. 2012). The connecting variable between presence and meaning is the emotional response to the experience, where a higher physiological activation strengthens the sense of presence. In turn, this means that it is possible for virtual simulations to make subjects feel a higher sense of presence than in real-world counterparts.

2.1.3 Breaking the presence

If the user is engaged in a VR application and to some extent is experiencing presence, then a detrimental event to the experience can be a *break-in-presence* moment. Just as the feeling of being present in the virtual environment disappears, the user acknowledges that she is in the real world, wearing a Head Mounted Display (see Section 2.3 for Head Mounted Display) and holding a couple of hand controllers. Such a moment can be the result of external factors, and in most cases might not be the fault of the application itself. People speaking, the user tripping on wires or when motion tracking of sensors is lost are examples of events where a break-in-presence moment can occur (Jerald 2016). This effect can be important to keep in mind during testing of the application with users, as interruptions by the supervisor or other spectators can result in such a *break-in-presence* moment. Especially when testing the application on conferences or fairs, where a lot of noise and other factors can affect the user testing the application, such break-in-presence moments can frequently happen.

2.1.4 Motion Sickness

It is not only a break-in-presence moment that can be detrimental to a virtual reality experience, *motion sickness* is one of the most common negative health effects of VR (Jerald 2016), and is found to be a larger problem than for normal desktop viewing (Sharples et al. 2008). Both *scene motion* and *vection* is known to induce motion sickness (Jerald 2016). Scene motion is the movement of the entire virtual environment that would normally not occur in the real world, while vection is an illusion of self-motion. Reducing such effects can then help to reduce motion sickness in users of such an application. As individuals repeatedly are exposed to VR, they will eventually get used to or adapt to these effects, but it does not happen immediately (Jerald 2016). AS far as possible, motion sickness should be avoided, but for a virtual internship it can be necessary to have elements that are known to induce motion sickness, in order to properly present the workplace. For example, in FiskeVR, the user can drive a boat themselves, which will include vection, but as this is an important part of the job it is necessary to include in the application. An alternative is to give the users the possibility to skip these parts, if they feel nauseous.

2.2 Human-Computer Interaction in VR

A lot of traditional Human-Computer Interaction design applies just as well to VR, but it additionally brings new challenges and opportunities for users to interact with an application. It is vital that the designers of the application effectively communicate how the virtual world and its tools work so that the users can achieve their goals (Jerald 2016).

2.2.1 Moving in the game world

The main problem with the movement of the player in the virtual world space is that it can result in motion sickness (as described in Section 2.1.4). There are three main ways of moving the player, listed below:

- **Walking**: The system will track the user's feet or position in some way so that the direction or speed (or both) are sensed and creates motion in the virtual scene. Usually, there are limits to the size of the walking area, but it can also be unlimited by, for example, using a specialized 360 degrees 3D mill that acts as input to the application.

- **Steering**: The player controls the viewpoint's speed and direction in some other way than walking, for example, with a joystick. Steering is useful when traveling greater distances or when transporting the player in a vehicle of some sort. This technique should be utilized with care, as it has a high probability to induce motion sickness in the users.

- **Teleporting**: Teleporting is technically a workaround for the motion sickness that steering can induce. When teleporting, the player is moved instantly, or very fast, to a chosen location in the scene. Usually implemented by letting the player point at an area with a ray or arc pointer, then do additional input like a button press to teleport to that location. Teleporting might be less realistic, but can help reduce the motion sickness in the user, and is a common practice in many VR games where the player is non-stationary, and the virtual world space is larger than the space tracked in the real world.

All three techniques have advantages and disadvantages. Walking is the most realistic locomotion technique by tracking the actual position of the user and mapping that to the simulated environment. The main drawback is that real-world objects and walls usually restrict the walkable space. Therefore, the walking technique is often combined with steering or teleporting, which allows the user to move further than the real world barriers allow. Steering has the obvious advantage that the player can move long distances quite realistically, being able to observe the travel and continuously adjust speed and or direction as they go. The obvious drawback is the higher chance in inducing motion sickness. Teleporting solves the problem of motion sickness that steering can introduce, but at the cost of realism and intuitiveness (people are not, in general, used to the concept of teleporting around in their environment). If the target group of the application is new to VR, teleporting can be a good technique to reduce the negative effect of motion sickness when they test the application. Ultimately, if possible, all techniques can be implemented, and the users can choose what suits them best.

2.2.2 Interacting with objects

Interacting with objects are done in a different way than regular desktop applications. Jerald (2016) lists a set of patterns; two main patterns are:

- **Hand pattern**: The user can touch the object with their hand (controller), mimicking real-world interaction. The user will have to physically reach out, and press a button on the controller to, for example, grab an object or flick a switch. The object needs to be in arms reach of the user in order for the interaction to be possible. The application should then make sure that objects are close enough, or give the user the ability to move objects towards them, or moving them closer to the objects.

- **Pointing Pattern**: This is used to select and interact with objects at a distance. Usually, a pointer (can mimic a laser pointer) extends from the hand (controller), or head (HMD) of the user and the first object it intersects with will be able to be interacted with in some way. This pattern can be used to pick up an manipulate objects at a distance or to chose options and browse a menu in an in-game user interface.

2.3 Virtual Reality Technology

There is a multitude of technologies and techniques to present a virtual environment for the user. First of all, a VR system, like most user-centric computer systems, consist of some components that let the user perform actions and receive feedback. A cycle between the user and the system, is depicted in Figure

2.3. Here the system will track input data from the user, like hand movements or button presses. The system interprets this input data, alters the digital environment accordingly, renders the environment, and finally displaying the environment to the user through a display (Jerald 2016). The biggest difference between a VR system and a regular desktop system lies in the display and the input technology that the system employs.

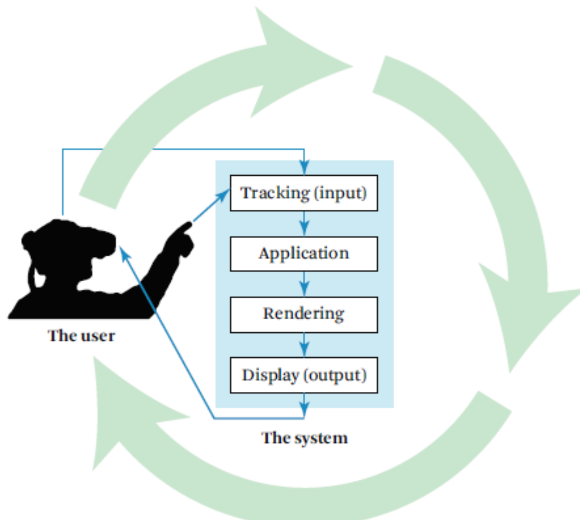


Figure 2.3: The continuous cycle of user-interaction and system feedback (Jerald 2016).

2.3.1 Displays

In general, three types of displays are used for virtual reality. All three types of displays bring their feats and caveats, and it is essential to choose the right for the use case and the developed application.

Head Mounted Display Known as an HMD, the Head Mounted Display is the type of display seen in modern commercial virtual reality systems, like the HTC Vive. These displays are as implied by their name, mounted to the head of the user. The application then tracks the position and orientation of the HMD, updating the display to give the appearance that objects in an environment stay stable (Jerald 2016). These displays can be very immersive for the user if implemented well in both hardware and software. This is the display technology that is most relevant to this thesis, as it gives the user the most immersive experience of the workplace.

World-Fixed Displays These displays render the graphics onto a real-world surface. A typical system that uses this technique is VR CAVE's, which is a system that surrounds the user by surfaces that have graphics rendered onto them. These systems are usually expensive and are set up for some specific use case. For example, Statoil (now Equinor) used this technology to do subsurface analysis, as a tool to look for oil (Hybertsen 1998).

Hand-held Displays Handheld displays are typically a smartphone or a tablet, and most frequently used for augmented reality applications. As smartphones have grown ever more powerful and sophisticated, AR on the phone has become more mainstream as a result. Another contributor to the popularity increase in AR on these devices is Apple and Google's commitment to creating tools that help developers develop these experiences, with ARKit and ARCore, respectively (Summerson 2018). There are VR experiences for hand-held devices as well, where the phone is mounted in a head mount, effectively turning the phone into an HMD.

2.3.2 Input

Like many other computer systems, VR systems can include buttons and triggers. What is more, differentiating is the ability to track the user's gestures, movements, and position.

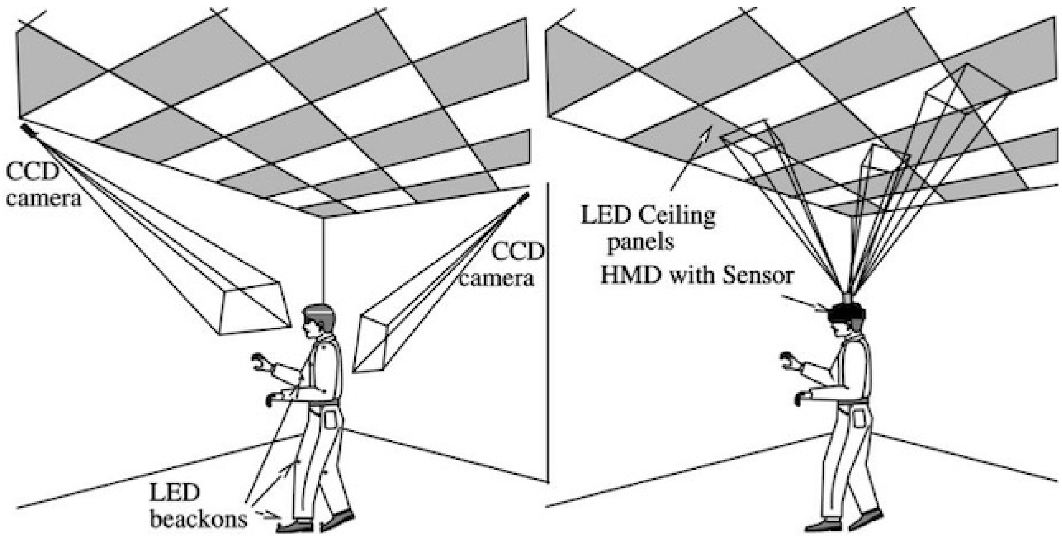


Figure 2.4: Two types of optical trackers. Left: outside-in tracking, right: inside-out tracking (Burdea and Coiffet 2003).

Position Trackers In 3D space, objects have six Degrees of Freedom (DoF) for movement. To accurately track the position of a user's hands or head with respect to the presented virtual world, all the axes of movement must be tracked. One widely used type of positional tracker is the optical tracker. An optical tracker is a non-contact position measurement device that uses optical sensing to determine the real-time position and orientation of an object (Burdea and Coiffet 2003). They use triangulation and requires a direct line of sight to work properly. Two different versions, *outside-in* and *inside-out*, of optical trackers can be seen in Figure 2.4. Position trackers are usually used to track the position of handheld controllers, the HMD and other objects in the VR play area. The most frequent technology used on commercial VR HMDs today is some version of inside-out tracking.

Controllers

The most used form of controllers in commercial VR systems is handheld controllers, usually with input buttons, a joystick or a touchpad, and 6 DoF position tracking technology. Both HTC Vive, Oculus Rift, and Windows MR uses this type of controllers; therefore, this is the technology deemed relevant for this project. More about hardware can be seen in Section 5.3.

2.4 User Interfaces in VR

Even though VR is three dimensional, it does not necessarily mean that traditional forms of UI are unnecessary. In the real world, humans are surrounded by 2D representations of content, in the form of screens. In a Google I/O talk from 2017, the designers talk about how they think of UI in VR. The design team approaches it as *screens* that convey information and/or can be interacted with (Google 2017). As in the real world, screens in VR have an intended viewing distance depicting the intended distance to view the UI, where it should be eligible. For example, a representation of a small smartwatch on the user's arm will have a smaller font size than for a large billboard placed in the game world.

2.4.1 The Distance-Independent Millimeter (dmm)

An established challenge in user interface design is the different screen densities from device to device. Event though screen size stays the same, the density of screens can vary, where density is pixels per area. In addition to density, screen sizes and aspect ratios vary, but that is another problem. To solve the screen density problem designers and developers usually apply a variant of a density-independent pixel measurement to avoid having to think about every possible density when designing the user interface. For VR the problem differs slightly, but are somewhat similar: how can the eligibility of the content in a screen be ensured at the intended viewing distance? Google solves this by defining a *distance-independent millimeter* or *dmm*, where 1 dmm is 1 mm at 1 m distance from the user, 2 mm at 2 m distance from the user and so forth. This way, the user interface can be designed in dmm size and include the intended viewing distance, so that at implementation time the size of the user interface is correct for the intended distance. An example of this can be seen in Figure 2.5.

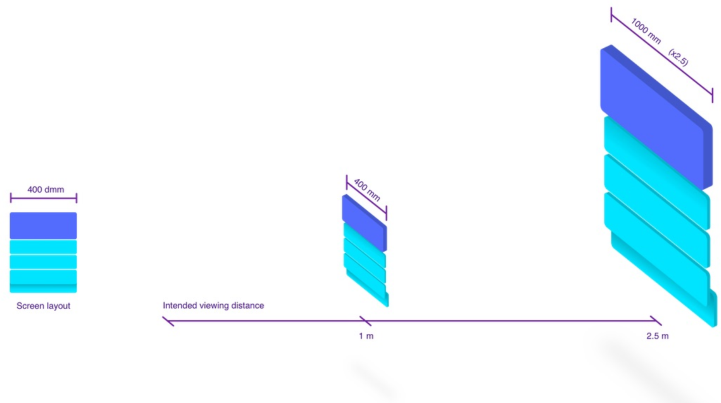


Figure 2.5: Layout scaling as intended viewing distance of the layout increases. The original specification is 400 dmm, so at 2.5 m distance, the layout is scaled to 1000 mm. Example taken from Google's VR Design Resources at `https://developers.google.com/vr/design/sticker-sheet`

2.4.2 Types of User Interfaces

Where do different parts of an interface go in VR? It is not necessarily ideal to reuse existing concepts from regular computer applications or games directly. In a computer game, two main *classes* of UI can be identified: *menus* and *heads up display (HUD)*.

Heads-up Display Traditional desktop or console games usually use some form of a heads-up display (HUD) to display menus, stats and information while in-game (see example in Figure 2.6a) as a 2D overlay to the scene. A HUD can be used in VR as well by connecting UI elements to the users head, making them follow head movements and rotation. However, this will not always be an ideal solution (Jerald 2016). An alternative is to place different parts of the HUD at the user's body or different areas relative to the user's body, or objects in the game. For example, the HUD elements can be placed on the user's arm, as shown in Figure 2.6b. This way, the HUD can be less

(a) Typical HUD in a computer game showing stats, map, objectives and available skills and items (screenshot from *Torchlight II*)

(b) HUD displayed on the users arm. From Leap Motions blog (`http://blog.leapmotion.com/wp-content/uploads/2014/12/arm-hud-arm.png`)

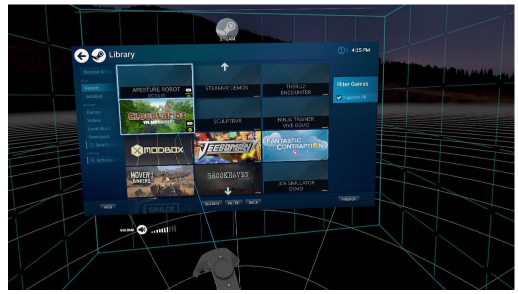
Figure 2.6: HUD in a typical computer game compared to a HUD on the user's arm in VR.

intrusive when wearing an HMD, and the user can bring up the HUD when wanted. Score or progress can also be presented relative to the location of the task or object related to what it represents, integrated into the environment. For example, a task can have a score level, or completion indicator hovering above the area for the task. This way the UI elements can be less intrusive, but still available and visible for the player.

Menus Usually, menus are optimized for being controlled with a keyboard and mouse or a game console controller. Figure 2.7a show a typical example of a menu in a game, with buttons laid out in a 2D fashion and presented to the user easily clickable or selectable with the mouse pointer. In VR the player will have no mouse pointer or keyboard[2], and menus need to be placed in 3D space, or *worldspace*, while still retaining both intuitive actions and readability for the user. The dmm specification (Section 2.4.1) can help focusing on readability

[2]Generally speaking. There are examples of applications that use a VR HMD, a mouse and a keyboard or gamepad).

(a) Typical 2D menu in a computer game, where buttons are clickable with the mouse (screenshot from *Shadow Tactics: Blades of the Shogun*).



(b) The SteamVR menu with HTC Vive controller visible. From `https://steamcdn-a.akamaihd.net/steam/apps/491380/ss_1fc77e21fef6c8f6d5e46169271bbe702e0e2900.1920x1080.jpg?t=1469807458`.

Figure 2.7: Menu user interfaces in regular computer games and virtual reality.

and eligibility when designing and implementing menus in 3D space. A typical example of a VR menu is depicted in Figure 2.7b, where the menu is placed at a certain distance from the user, and the Point Pattern (Section 2.2.2) is used to select buttons and to scroll in lists. This way, a lot of existing metaphors from regular computer software can be reused, making a familiar approach for many users.

2.5 Games

It is necessary to focus on game research and development when researching and developing virtual reality applications, as this industry influences the VR industry heavily and the two areas have many parallels (Zyda 2005). Games can be immersive, captivating experiences. James Paul Gee (2003) states that good games are *learning machines*. They get themselves learned and learned well (Gee 2003). He further states that schools, workplaces, and academia has a lot to acquire from games when it comes to learning. Games are learning machines if they are *good*, but what makes a game good? Good, compelling games do not happen by accident, just like other gripping media, but use strategies and techniques to craft a unique emotional experience for the user (Isbister 2016). Good games tend to give information on demand and just in time, not out of context and all at once. Information is spread throughout the game world, making it clear how it applies to that world (Gee 2003). A virtual internship is not a regular game; its primary goal is not to entertain, but to inform. The fact that it is not a regular game does not mean that borrowing concepts from traditional video games are not a viable thing to do, but it could be that it must be done with more care, thoughtfully choosing what concepts and techniques provide value for such an application.

2.5.1 Serious games

A virtual internship can be defined as a *serious game*: a game with a primary function other than entertainment (Wouters, Van der Spek, and Van Oostendorp 2009). A game can be defined to have three components: software, art, and story. A serious game has an additional component, *pedagogy* (Zyda 2005).

Serious games have a set of learning outcomes, which can be divided into four main categories: *cognitive*, *motor skills*, *affective*, and *communicative*. Cognitive outcomes include knowledge (declarative and procedural) and skills (problem-solving, decision making, and situational awareness). The second outcome, motor skills, requires procedural knowledge, knowing how to do something, then practicing this behavior to improve and make it less error-prone. Affective outcomes include attitude (internal state influencing choices and actions) and motivation (a prerequisite for real learning to commence). The last outcome is communicative, which includes social skills, cooperation, and communication. Depending on the workplace or work situation simulated, the important learning outcomes might vary. The difference in learning outcomes is, to some extent, the real difference between a virtual internship and a workplace training application. A workplace training application's learning outcomes will vary on what kind of work it is depicting: physical work will have motor skills, service work would have communicative skills, etc. In a virtual internship, however, these outcomes will merely be a side effect, while the main outcome will be *affective* and to some extent, *cognitive*. A virtual internship is not made for employees to learn how to do something; it is made to change people's internal state or feelings towards a career, influencing career choice and even motivating towards applying for jobs. Reflecting on the purpose of the serious game throughout the design is essential, to avoid incohesive and conflicting games (Mitgutsch and Alvarado 2012). For a virtual internship, it will be equally important to demotivate people from careers as motivating, since choosing the wrong career can result in dissatisfaction, and even dropping out of education when an individual realizes that they have chosen entirely wrong. It does not mean that these applications should be dreadful or demotivating in any way, but they should be realistic enough so that the users understand the foundations of the workplace. Overusing concepts from video games might be detrimental to the effect of these applications, as the amount of "fun" users feel are completely out of proportions with what they would feel in the job. In, for example, a game for learning youngsters math concepts, techniques can be used to boost the amount of fun and curiosity the user feels immensely, in order to motivate towards learning math concepts. In a virtual internship, the goal is a little different, the user should get an impression, and the application should motivate towards either exploring the work more or realize that this is not the correct path for a career.

Characteristic	Techniques
Challenge	Goal
	Uncertain Outcome
Fantasy	Intrinsic fantasy
	Extrinsic fantasy
Curiosity	Sensory curiosity
	Cognitive Curiosity

Table 2.1: The characteristics that makes learning in games fun (Malone 1980).

2.5.2 What is so special about games?

It is useful to investigate some of the main concepts that games use to keep players engaged and motivated to learn. Think about the amount of information a teenager willingly learns, remembers, and employs in practical usage when playing a game like World of Warcraft. They learn themselves a whole new world, a system describing characteristics of their character, traits held by items, social interaction systems, maps, places, certain actions that have to be done in order to accomplish and gain items and experience and so on. These are vast worlds with tons of information, where people willingly spend hundreds of hours learning and mastering, in addition to paying a monthly fee for accessing it! The world these games present supply motivation in some way that makes it seem worthwhile to master. Malone (1980) defined three categories to induce enjoyment in instructional video games: *challenge*, *curiosity* and *fantasy* (see Table 2.1).

Challenge is the fact that the game needs a goal or several goals (tasks) that can be completed. It also introduces the notion of an uncertain outcome, where the player should not know whether they will be able to complete these tasks or not. This is where a virtual internship and a regular game might divide, the notion of the player being uncertain and feeling motivated to beat a challenge might not be as important, what is important is that they understand the tasks themselves and gain insight. Additionally, the player should receive feedback on their progress towards the goals and completion of them. Introducing multiple level goals, or meta-goals can let players adjust the difficulty themselves. Multiple level goals are additional goals that can be completed, and that might be more difficult or harder to attain. Meta-goals are goals that enhance already existing goals. For example, completing a particular goal, but within a specific time frame, with a particular score or in a certain way so that the player can earn even more points in the game.

The second category, fantasy, is important for a virtual internship, though the fantasy should not be something out of fiction. The two main types of fantasies are *intrinsic* and *extrinsic*. Intrinsic fantasies are fantasies where the challenges of the game are presented in terms of the world in the game (Malone 1980). For example, a football game uses an intrinsic fantasy: the fantasy world of the game of football, and the task is to play football and win the match. On the other hand, if the goal is not directly connected to the tasks of the game world, it is deemed an extrinsic fantasy. For example, a typical quiz game where, as the user answers correctly, they get closer to some fantasy goal not related to the questions. A virtual internship will have an intrinsic fantasy, where the fantasy is the workplace, and the tasks are directly related to the work there.

The third category, curiosity, is an important factor in motivating the players to learn and explore, with two subcategories: sensory and cognitive curiosity. Sensory curiosity can be aroused with vivid visual and audio effects, enhancing the fantasy it presents. The sensory curiosity is valuable in a virtual internship since the *workplace fantasy* can be enhanced with the use of graphical effects and suitable sounds to give the user a more immersive experience. The use of advanced enough graphical effects can be even more important for a virtual internship than regular games in this regard, as the representation of the fantasy (in a correct manner) is an important feature. Besides, graphics and sound can be used as a reward for completing tasks or conveying information more effectively than regular textual info. Cognitive curiosity is "a desire to bring better knowledge to ones knowledge structures" (Malone 1980). People want to complete their void of information about a topic if they know that one exists. Lowenstein (1994) refers to the works of Jean Piaget, a Swiss psychologist that saw curiosity as "the product of cognitive disequilibrium evoked by the child's attempt to assimilate new information into existing cognitive structures" (Lowenstein 1994). Since Piaget dedicated a lot of his work to child development, the statement reads as such, but it is still valid for any human being in some sense. Humans want to bring stability to their own perceived cognitive disequilibrium; therefore, they seek new information on areas where they feel this is needed. This type of curiosity can be hard to induce in players specifically. One way could be to give the player enough information about what to do but leave some room for experimentation and further investigation to the player themselves. Telling the player exactly what to do and when to do it, might reduce their experienced cognitive curiosity when playing.

2.5.3 The emotional aspect

Providing a challenge, a fantasy and inducing curiosity is good pillars of a game, but many scholars look at games as possessing a unique ability to induce different emotions and create emotional experiences on another level than almost any other media (Isbister 2016), with two fundamental capabilities setting games apart from other media being *choice* and *flow*. Choices in the game should be interesting and make the player feel like they have an impact on the game world. Choices can be moment-to-moment choices like what actions to take in a single situation that will directly impact the outcome of a task or high-level choices in the storyline that has a wider impact on the game world. Flow, the theory of *optimal experience* by Csikszentmihalyi describes that the best moments of life usually occur when a person's body or mind has to stretch to its limits in a voluntary effort to accomplish something difficult and worthwhile (Csikszentmihalyi 2014). The Flow theory includes eight elements that contribute to this experience:

1. There is a task that requires skill that we can complete. Both challenge and skill must exceed some threshold.
2. Must be able to concentrate on the task (what we are doing)
3. The task has clear goals
4. The task provides immediate feedback
5. Deep, effortless involvement, removing the worries and frustrations of everyday life
6. Have a sense of control over actions
7. Concern for the self disappears, yet after the experience, the sense of self emerges stronger
8. The sense of time is altered

When a game offers the player impact-full choices and keep them in flow, it can induce a new set of emotions in the player; *social emotions* like empathy, affection, camaraderie, grief or sadness (Isbister 2016). Building on the Flow Theory, Sweetser and Wyeth (2005), created what they termed GameFlow which builds on the eight elements of flow to create eight elements more directly connected to games: *concentration*, *challenge*, *player skills*, *control*, *clear goals*, *feedback*, *immersion* and *social interaction* (Sweetser and Wyeth 2005):

1. **Concentration**: The amount of stimuli provided to the player should be sufficient, and quickly grab the players attention and focus throughout the game. There should be a workload that is high, but not overly high, without burdening the players with unimportant tasks.
2. **Challenge**: Games should have challenging gameplay, matching the player's skill level and providing possibilities for the player to vary the

difficulty.

3. **Player Skills**: The game should provide an opportunity for the players to evolve their skills and give them a sense of achievement. A manual should not be needed to play a game, and lengthy explanations will bore the player.

4. **Control**: Players should feel a sense of control over movement, interaction, interface, and game world. When errors are made, they should be recoverable. Players should feel a sense of control over what performed actions and how to play the game, not as if they are following along a strict path laid out by the game developers.

5. **Clear Goals**: The goals of the game should be presented at appropriate times and be clear.

6. **Feedback**: The status of the progress towards goals should be known, and actions should result in immediate feedback.

7. **Immersion** The game should make the players less aware of their surroundings, become less self-aware and worry less about everyday life. They should be involved in the game, both emotionally and viscerally.

8. **Social Interaction**: Creating opportunities for social interaction and competition between players.

This model is not necessarily suited for game developers to evaluate the quality of the game they have made directly, but can be useful as a tool for identifying issues and understanding player enjoyment in games by using the model as a base for creating user tests or identifying issues (Sweetser and Wyeth 2005). So in itself, the model is not necessarily useful to evaluate a game directly on whether it is enjoyable or not, but the elements identified are important to give the developer a better understanding of what makes a game enjoyable, identify issues and employ techniques to achieve the different elements of the model.

This model, to induce emotions in the player is all well and good, but it is imperative to employ these techniques with much greater care and less extensively in a virtual internship than in a typical game. They can be used, but selectively. For example, the amount of challenge, skills, and reward for completing tasks can be reduced in a virtual internship as the focus is not this optimal state of working hard towards completing specific tasks, but instead learn about these typical tasks that will need to be done in the real world workplace. These tasks can even be boring ones that are necessary for the represented particular line of work, as they might be a big part of workers' everyday life. Clear goals, feedback, and immersion are fundamental aspects, as these will make the tasks in the workplace clearer and make the workplace simulation more realistic and immersive. The players should feel like they are in a building lot or on top of a wind turbine, not in a room with an HMD on, connected to a computer. The element of social interaction and its importance can vary significantly from workplace to workplace and is something that might be relevant to research in future applications where cooperation and interaction between workers, or customers, are primary elements.

2.6 Career guidance

The globalization of the world, combined with new occupations created with new technology, has vastly increased the number of choices that young people have when choosing a career. People want to have jobs where the tasks interest them, but few will know what workers do on the job (Gottfredson 2005). NAV organizes excursions, visits and lets people try out specific jobs in periods to give people a chance of understanding what tasks are frequent, and if it interests them. This can be a costly process if individuals have to visit countless companies without results. The experience of VR combined with the possibilities that game design and technology provides, could lead to entirely new ways of recruiting and guiding people towards a choice of career. Instead of visiting and trying out many jobs in real life, they could try out every thinkable job in VR, maybe even jobs they did not think about as an opportunity. More accurately customized measures for getting an individual back into the job market could be employed, by first reducing and more clearly defining the scope of interests and possible careers for that individual, through the use of virtual internships. Companies themselves are starting to employ this technology in recruitment efforts: Jaguar is building recruitment games, Deutsche Bahn is using it at career fares to overcome their aging workforce, The British Army lets people drive tanks in VR, and Commonwealth Bank of Australia use VR to test candidate's decision-making skills (McLaren 2017). It is in the interest of the companies, as much as the employees, that the workers are correctly employed into a job they will enjoy, thrive, and be productive. The connection to the industry when creating VR experiences that represent a workplace is essential, to make it as real and representative as possible. A prominent part of a workplace that cannot necessarily be represented is the social system, social environment, and inter-relationship between workers. It could be represented, but then not only all relevant occupations would have to be represented, but every single physical workplace with its unique social elements. Also, they would have to be updated if new leadership or much new staff is employed, resulting in an unfeasible project. Instead, it is vital that the representations give the candidates a general impression of the tasks and nature of the work, then if they are interested, further steps like a visit to a company or an internship can be evaluated. A VR application developed to represent and inform an individual will not be of as much use to unemployed older individuals that might already have their career selected, but are unemployed for other reasons. The scope of the target group of such an application will also be not just unemployed people in public employment service programs, but students and pupils at younger ages that might not have taken choices in careers yet. Using games for career guidance could help these groups, they are used to games, and games and technology are an increasing part of peoples childhood and youth.

2.7 Creating a virtual workplace

2.7.1 The development process

VR applications are complex systems that require knowledge in multiple disciplines. A VR application has three essential requirements that characterize and differentiate them from other software systems (Seo and Kim 2002):

- Real-time performance while maintaining realism and presence.
- Modeling objects and their physical properties, function, and behavior.
- Consideration for the different interaction concepts when completing different tasks and using different input-output devices.

Further, Seo and Kim (2002) propose a structural approach to deal with these specific challenges. The model developed is called the CLEVR spiral model and can be seen in Figure 2.8.

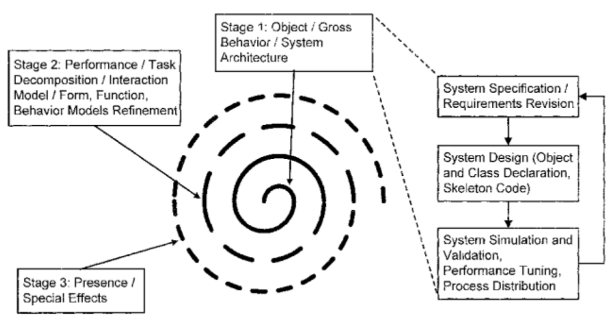


Figure 2.8: VR development model CLEVR, from (Seo and Kim 2002).

Through this approach, the objects and overall system behavior is first created, then performance, form, and function is implemented. Finally, special effects and graphical improvements are applied. The general thought behind the model is good, but development should not only be incremental. What is important is that this technique might be very well suited for specific components in a VR application, while the overall process of development should be both incremental and iterative. An incremental process aims to deliver subsystems or parts of a complete system, entirely developed and tested, without the need for revisiting in subsequent iterations (Cohn 2014). An iterative process plans for improvement of features in future iterations, therefore the process should both be incremental and iterative when developing a VR application.

2.7.2 Simulating a workplace

No project similar to the virtual internship project was found in existing literature, but there are several examples where VR technology is used to simulate situations and tasks on a workplace for training purposes. Virtual reality offers limitless possibilities in training, simulation, and education (Kizil and Joy 2001). VR can be a cost-effective alternative solution to train new workers by reducing both time and money needed for training (Ayala García et al. 2016). Workplace training is not the goal of this project, but simulating a workplace for training on the tasks versus simulating a workplace in order to learn about it and the tasks, does overlap, as the most significant difference is the fidelity required. A workplace training application will need to be more accurate than a virtual internship regarding the actual execution of the tasks. For a virtual internship, it will be adequate with tasks that give the user an impression of the workplace; they do not need to evolve muscle memory or experience.

Aside from economic benefit, VR simulations can be flexible and modified for new requirements, simulate a wide variety of scenarios, be multilingual, and it is easier to track and analyze the training (Wyk and Villiers 2009). A virtual internship will benefit from these points, just as a workplace training application will. By being flexible and modifiable, the virtual internship can be changed if tasks, rules, or laws change for specific workplaces.

The ability to simulate a wide variety of scenarios could enable these virtual internships to simulate situations and parts of a job that might not even be available if people interned there in real-life. If a job seeker gets to intern at a company, it could be that they cannot access certain areas or complete specific tasks because of lack of training, or security and safety. With a virtual internship, these parts of a workplace can be safely tried out by anyone curious, even simulating situations where health and safety would be on the line. For some workplaces, for example, emergency services VR simulations enable untrained curious job seekers to try to drive a police car fast through traffic or save a trapped human in a burning building. The point is that VR enables job seekers to try parts of a job, that might be inaccessible otherwise.

The ability to make the virtual internship multilingual can help immigrants who use NAV services by making the application more accessible to them. In addition to being multilingual, these applications can include functionality for explaining items or elements in the application more clearly and detailed, even through voice acting, which can help both people with learning disabilities and individuals that are just learning the local language.

Chapter 3

Method

This chapter presents the research approach, with methods for data generation, data analysis, evaluation, and development process.

3.1 Research Approach

A research project consists of specific methods, activities, and actions in order to generate, collect, and analyze data to reach an informed conclusion. The methods and actions for this thesis are selected based on the nature of the research, but also relies on available resources. The thesis research strategy is divided into two: one case study and one design and creation process. First, a case study will be conducted with the existing virtual internship application *FiskeVR*, and then a design and creation process will follow, with development of a new virtual internship. Figure 3.1 depicts an overview of the strategies, data generation methods and data analysis methods employed in this thesis.

3.1.1 A case study of the existing Virtual Internship application

A case study focuses on an instance of the *thing* that is investigated (Oates 2006). The case study in this thesis is performed on the already existing vir-

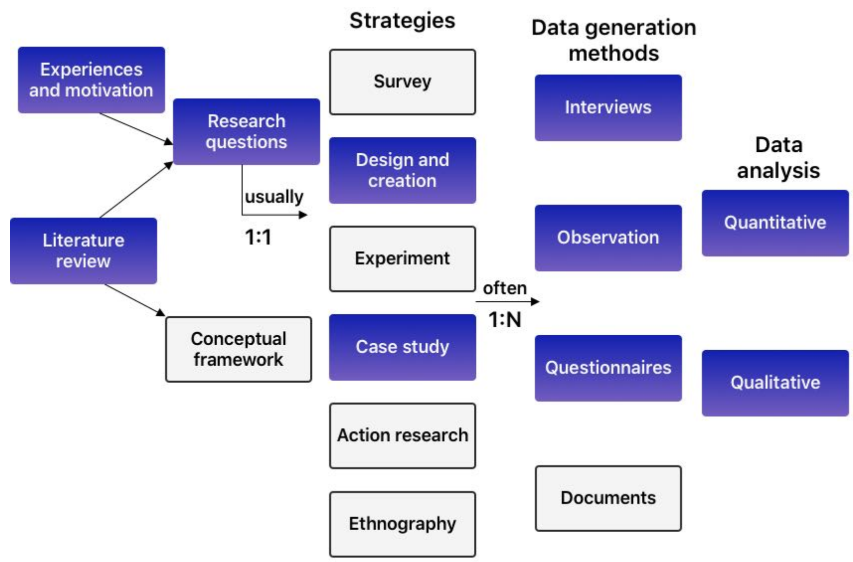


Figure 3.1: Oates research model describing methods and activities, adapted from (Oates 2006). Blue boxes are the ones applied to this thesis.

tual internship in order to understand the topic, and more clearly define the subsequent development of a new virtual internship. This type of case study is referred to as an *exploratory study* (Oates 2006), and is especially fitting when the topic or specific case is not covered as much in the available literature. Therefore, a case study on the existing virtual internship was deemed necessary, to be able to understand and form a picture of the topic where not everything could be found in the literature. The choice of what virtual internship to study is simple, as *FiskeVR* is the only already developed. The data generation methods used in the case study is observations (see Section 3.2.3), questionnaires (see Section 3.2.2), and interview (see Section3.2.4).

The case study was mainly performed in the fall of 2018 before the Design and Creation study described in Section 3.1.2 as part of a pre-research project to this thesis, referred to as course TDT4501 at NTNU. Both results in the case study, interpretations of them and some literature is from the report in this subject. There are however both additions and subtractions to the study compared to that report, and the case study is an essential part to build knowledge on virtual internships.

Philosophical Paradigm

The philosophical foundation of the case study is interpretivism. In interpretivism, there is no single truth, and findings and knowledge are constructs of our minds (Oates 2006). Interpretivism has, in general, a strong preference for qualitative data, but in this study, both will be used as a tool to uncover important aspects of features and usage of the application. Often, an expectation in interpretivism is that there will be no single explanation to what occurs, but presented explanations are based on the evidence the data provides.

Data Generation

In Table 3.1 all dates where tests where executed and data gathered is presented.

Table 3.1: Dates, location, users and data generation methods for the case study on FiskeVR.

Date	**Place**	**Users**	**Methods**
10th September 2018	VR Lab Dragvoll	6 NAV users, 1 NAV Employee	Questionnaire, Observation
20-21 September	Forskningsdagene Frøya	7 pupils (aged 15-17)	Questionnaire, Observation
28 September 2018	Researcher's Night	16 pupils (aged 16-18)	Questionnaire, Observation
30 October 2018	VR Lab Dragvoll	20 NAV users, 5 NAV employees	Questionnaire, Observation, Interview
26 November 2018	VR Lab Dragvoll	13 NAV employees	Questionnaire, Observation

3.1.2 The Design and creation of a new virtual internship

This strategy focuses on research that will develop something, called *artefacts*. It uses an iterative process where learning through making is an important aspect (Oates 2006). The artifact in this thesis will be an *instantiation*: a working system that demonstrates that constructs, models, methods, ideas, genres or theories can be implemented in a computer-based system. This is a well fitting method for the development of a new virtual internship, where knowledge while making is important since many aspects of such an application are not yet fully realized.

Design and creation employ a development methodology or process. The process followed in this thesis is an iterative agile process using kanban for each iteration to keep control of requirements to implement and test. The iterations in the development consist of three main iterations, with two sub-iterations for each main iteration. The methodology can be seen in Figure 3.2. The duration of a major iteration is approximately one month, while a sub-iteration is two weeks. A sub-iteration consists of development and a report. The report is sent to important stakeholders in the project, including TrønderEnergi and NAV, to receive immediate feedback and fine-tune the implementation underway. The report consisted of the main implemented features for that sub-iteration and a recorded video of the application, demonstrating the new features. For each main iteration, it is conducted a user test, where data is gathered, analyzed and evaluated to bring new perspectives, ideas and get knowledge on problems and features that are needed for future development. This knowledge is then used as input for the next iteration.

Just as other strategies, design and creation involve data generation methods. The ones applied for this process are observations (see Section 3.2.3) and questionnaires (see Section 3.2.2). For each main iteration, a user test is conducted, where the users answer a questionnaire. The author also observes

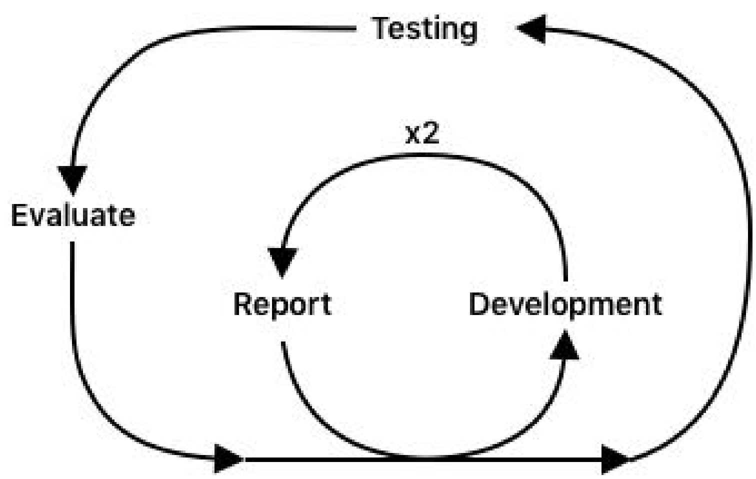


Figure 3.2: The development methodology used in the design and creation process.

the tests to gain further understanding of problems and fuel new ideas. The testers for each iteration vary, but all fall into one of the groups described in Section 3.2.1.

The main iterations is described in separate chapters (Chapter 6, Chapter 7, and Chapter 8) which each will describe implemented features, the process of implementation, testing and the results.

Data Generation

In Table 3.2 all dates where tests where executed and data gathered for WindturbineVR is presented.

Table 3.2: Dates, location, users and data generation methods for the design and creation of the new virtual internship.

Date	**Place**	**Users**	**Methods**
14 February 2019	VR Lab Dragvoll	33 students	Questionnaire, Observation
15 February 2019	VR Lab Dragvoll	11 students	Questionnaire, Observation
11 March 2019	VR Lab Dragvoll	5 NAV employees	Questionnaire, Observation
15 March 2019	VR Lab Dragvoll	5 NAV users, 3 NAV employees	Questionnaire, Observation
20 March 2019	VR Lab Dragvoll	6 pupils (aged 17-18)	Questionnaire, Observation
30 April	Bessakerfjellet Wind Park	2 wind turbine technicians	Observation, Interview
28th and 29th of May	Jobbhuset NAV Falkenborg, Trondheim	8 NAV users, 4 NAV employees	Questionnaire

3.2 Methods for Data Generation

This section presents the data generation methods.

3.2.1 Selection

The main target group of these applications is, in general, any user that would need career advice, or want to test out a specific workplace (unemployed or not). In this group of potential users, three groups have been tested: NAV users, pupils, and students. These three groups will be in slightly different places in their life: NAV users are generally unemployed (because of several reasons), pupils may not yet have chosen their career path while students have taken a more rigorous choice in what they want to do. Here, pupils are referring to people in lower level schools (year 8-13), while students are at college or university level.

In addition to the target groups of the application, two more groups have tested it, each with different stakes in the use and development: NAV employees and workers in the profession. NAV employees will use these applications as tools with the NAV users they are responsible for, and it is important that they both understand and partake in the development of the application. They will be facilitators for their users and will need to include the usage of these applications in their routines when guiding people towards a career.

Workers in the actual profession (in this case, wind turbine technicians), will be able to give valuable feedback and validate the realism of the representation of the job in virtual reality. TrønderEnergi made the technicians available for testing.

3.2.2 Questionnaire

All the statements use the Likert 5-point scaling (McLeod 2008) in the questionnaire. It requests the user to rate a statement from 1-5. There are two scales applied to this number range in the questionnaires, which can be seen in Table 3.3. The *rating scale* is applied to questions that ask the informant to rate an aspect of the application. The *agreement scale* is applied when the informant is asked to supply in what degree he or she agrees to a particular statement. Along with these types of questions, some long-form text answers are also used to let the users give more comments, feedback, and suggestions for improvement.

Rating	**Rating scale**	**Agreement scale**
1	Bad	Strongly Disagree
2	Inadequate	Disagree
3	Fair	Undecided
4	Good	Agree
5	Great	Strongly Agree

Table 3.3: The Likert scale used in the questionnaires and how the two different scale relate to the scale numbers.

3.2.3 Observations

Observations are useful to identify how and what people are doing, compared to what they say they do (Oates 2006). The observations done in these studies are unsystematic observations, where general usage, problems, and spontaneous feedback from the user is observed and noted, both as they use the application and afterward.

3.2.4 Interview

An interview is a conversation or discussion, usually with an agenda or specific (Oates 2006). During the research both structured and unstructured interview has been performed.

3.3 Data analysis

The data generated in both the case study and the design and creation study will both be quantitative and qualitative. The questionnaires generate both quantitative and qualitative data, while the interviews and observations generate qualitative data.

3.3.1 Quantitative

Quantitative data is data based on numbers. The analysis of them will interpret these data, look for patterns, and draw conclusions (Oates 2006). To find these patterns, the data must be organized, then either statistical methods can be applied, or the data can be represented in such a way that patterns can be seen. The quantitative data collected is *nominal* and *ordinal* data.

3.3.2 Qualitative

Qualitative data is all the non-numeric data generated by the data generation methods. The interviews, questionnaires (textual questions), and observations will all generate qualitative data, mainly textual data (words).

3.4 Evaluation

This section will first present a general evaluation, before a more specific evaluation of each study is presented.

The fact that the users get to go to the VR Lab and test out "cool" new technology can impact their understanding and perceptions of the application. Many subjects that test the application might be using virtual reality technology for the first time, which can implicate the result in both directions:

1. The users can be impressed by the technology itself, which can impact the results so that the users rate the application more favorably than they would do if the technology were well known to them.

2. The users can be overwhelmed by the technology, the controllers, and

the new ways of interacting with a computer system resulting in a more negative focus when rating the application.

When testing, both of these situations were observed: Some users are amazed of the fact that they can move around, interact with objects and be immersed in a new world, while others are completely overwhelmed by all the buttons and the new concepts of interaction. It does not mean that results should be dismissed, however, if these applications should be deployed they should be made accessible to as many as possible, therefore it is important to understand challenges different types of users face when using them.

Another important aspect is the principle of interaction between the researcher and the studied subjects (Klein and Myers 1999). The studied subjects are as much interpreters and analysts as the researcher, and they will alter their perceptions and horizon as they are introduced to new concepts. This effect would be smaller if the researcher never interacted with the subjects, but in this case, the author has supervised most tests the data originate from. The interaction between the researcher and the tests subjects can impact (and most likely will) the results, and it is important to be aware of this.

3.4.1 Questionnaires

People are asked to rate the realism and correct representation of the work, but does not necessarily have any experience or qualification to do so. In addition the questionnaires does follow the template that the virtual internship project uses, as a whole. This could affect the results in such a way that pressing matters not already uncovered are not thought about.

3.4.2 Case Study

A case study should occur in a natural setting, not in an artificial situation. In this case, the natural setting was not available as these applications are not yet employed for usage by NAV, so users had to come to the VR Lab at Dragvoll to test the application. Some tests occurred at career fairs, and other events, which in some regard can be seen as a natural setting since a use case for these applications could be on career fairs. Most data generation is, however, not in the natural setting. A problem with this is that it does not focus on how these applications would be received when used as tools by NAV employees for career guidance, it just focuses on the user, and how they act and react when using the application. As described earlier, the relationship (or just the presence) of the researcher can impact the results of this study.

When observations are conducted, the researcher should not interact or guide the subject, but this had to be done, as for many tests the author has been alone, both supervising and observing. In a team of several researchers this could be avoided by having a silent spectator as the observer, which would most likely also gather more data as they would not have to concentrate about both observing and supervising.

3.4.3 Design and Creation

The development of a new virtual internship follows a design and creation process, as earlier described (Section 3.1.2). During this process two main aspects are continually evaluated by the author: the process of the development and recreation of the workplace, and the usage of a new SDK for development (Virtual Reality Toolkit or VRTK) of VR applications. Since the author has no earlier experience in VR development, and very little experience in game development in general, the conclusions on whether VRTK is a valid and useful SDK, and the description of the process is not built on years of personal experience. It could be that for a seasoned VR developer, VRTK would not be necessary at all, and that the resulting implementation would be more solid if they used a different framework with less bundled code and examples. The danger is that the reason VRTK shows promise in this thesis, is that it is a framework well suited to someone new to development of games and virtual reality applications.

The fact that the process from real-world to a virtual environment seems complicated, could very well be of the same reason as mentioned above: little experience. As the author has no extensive experience in modeling, texturing and 3D design, it might feel more necessary to describe this process in such detail. However, it is still important, since a virtual internship is no normal game, it is representing something real and need to be correct and realistic.

In the final evaluation test, where NAV employees supervised the test, no observation was performed. This was a mistake, and should have been performed in order to collect more data than just from questionnaires, on how the user reacted to the applications when they were tested in a more natural environment.

In addition, as the development unravels it shows that the scope of this thesis becomes quite wide. By using a new development foundation, creating a new virtual internship, introducing a new menu system and several other new concepts it can be hard to identify which changes gives what results. Instead of identifying with security that, for example, the menu system implemented in this thesis is superior to the menu system in the other virtual internships, the result is that with some level of security we can identify that one of the things that is different impacted the result. By changing so many aspects at the same time it is hard to securely identify which changes are for the good, and which should be refrained from. One can maybe say that the results in this study takes many small steps in a lot of areas, but not huge steps in one specific area.

Chapter 4

Case study of an existing application

This case study consists of three main parts. First, a description and analysis of the current virtual internship application, FiskeVR, will be given. Then results from a series of user tests on this application is presented and briefly analyzed, and finally, a discussion of the results is performed. The case study is performed to gain a fundamental understanding of the project's current state and how these applications work and how they are received and approached by users.

4.1 FiskeVR

This section presents a description and analysis of FiskeVR. The application puts the user at a breeding and processing facility for salmon. This is the only virtual internship application developed, and it is useful to study and reason about features and choices done here to iterate upon for the next virtual internship. All tasks and main functionality will be described as briefly as possible. Each task contains two main features: a 360-video of the process in real life, and an implementation of the task that the users can try themselves.

4.1.1 Feeding

Feeding the salmon is done remotely in a control room on-shore. At this station, the user can see a 360-video from the real control room where the operator oversees and remotely controls the supply of food to the cages. Additionally, the user can move around in the digital representation of the control room and control the supply of food themselves. It is important that the salmon get enough food, but not too much as this will be a waste of money for the company. An overview of the control room can be seen in Figure 4.1a.

(a) The 360-video screen, textual description and control station in the feeding control room.

(b) The control station in the feeding control room.

Figure 4.1: The control room where the feeding of salmon is remotely surveyed and controlled by an operator.

In Figure 4.1b the computer the player can try to feed the salmons themselves. The computer controls the camera (which cage the user is watching) and the flow of food with a set of commands issued by the buttons at the control panel:

1. Numbered buttons: The numbered buttons choose which cage to watch on the screen.
2. "Opp" and "Ned": Whether to see the currently selected cage above or beneath the ocean surface.
3. "Start" and "Stopp": Starts or stops the flow of food to the currently

selected cage. The screen displays the flow of food into the cases, and when it starts to pass through the steam of salmon, it should be stopped.

4.1.2 Driving the boat

To get to the other parts of the application, the user will have to drive a boat that is anchored to the dock next to the control room. The controls of the boat (Figure 4.2a) mimics the controls of a real-life boat. It has an ignition, a lever throttle, and a steering wheel. The ignition is activated by the user moving the controller near the switch and pulling the trigger. The throttle is controlled by moving the controller near it, then dragging while holding the trigger button down (forward for forward throttle and backward for reverse throttle). The steering wheel is controller with the controllers touch-pad: touching the left side will steer left, and vice versa. There are two other main destinations, besides the control room in the game: the cages (Section 4.1.3) and the processing facility (Section 4.1.4). Green destination markers hover over these locations to guide the user when driving the boat (see Figure 4.2b).

(a) The boat controls: steering wheel, ignition and throttle.

(b) Destination markers shows destinations, when the user is driving the boat.

Figure 4.2: The boat used to transport the user between the control room, cages and processing facility.

4.1.3 The cages

The cages are placed in the sea, between the control room and the processing facility. This is where the salmon lives until they are large enough to be slaughtered and processed. One of the cages contains a 360-video of such a cage in real life. Additionally, there is a task to fix a rope on the cage, which is either damaged or missing.

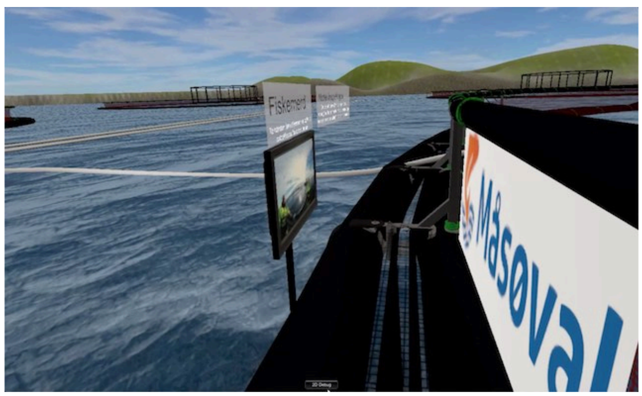
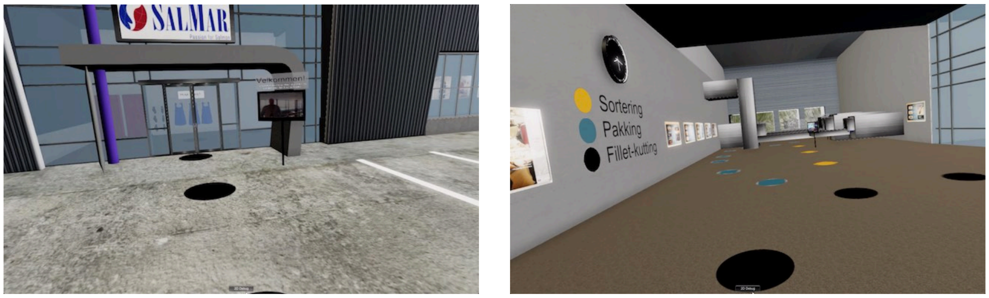


Figure 4.3: The user standing on the side of the salmon cage. The 360-video screen is seen to the left.

4.1.4 Processing Facility

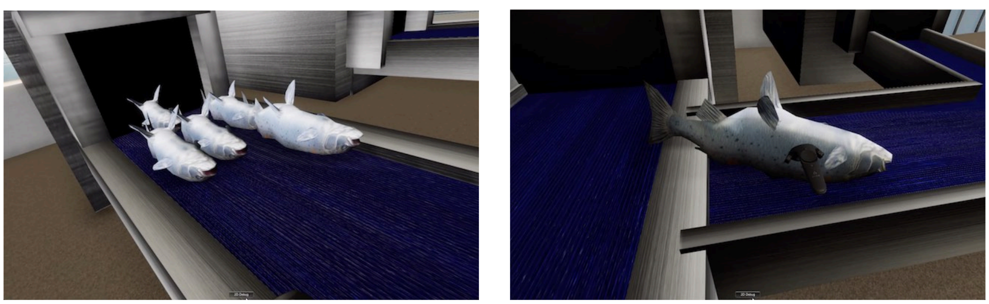
The user approaches the processing facility by boat and will have to dock the boat, go ashore, and enter through the main door. In Figure 4.4a, the exterior of the facility can be seen, with a video screen where the user can see the facility manager wlecome them to the facility. In the main hall inside (Figure 4.4b) the user can follow each of the colored paths on the floor to reach the different tasks that can be performed: sorting, packing and filet cutting.



(a) The exterior of the fish processing facility.

(b) The interior of the processing facility. The markers on the floor correspond with the ones on the wall, leading the user to the tasks.

Figure 4.4: The processing facility where the user learn what happens to the fish after slaughtering.



(a) The conveyor belt all the fish arrive at. The user needs to identify wounded fish by their pink marks.

(b) The conveyor belt where the wounded fish needs to be placed.

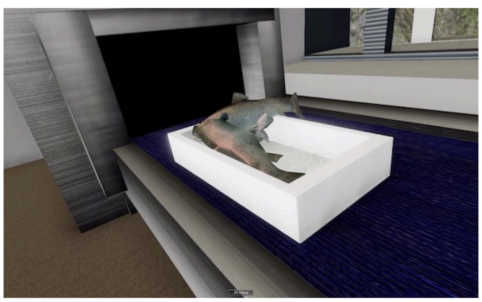
Figure 4.5: The sorting task, where wounded fish needs to be identified and sorted out.

Sorting

The sorting station is reached by following the orange markers seen in Figure 4.4b. The goal of this station is to identify wounded fish arriving at a conveyor belt, then sort these out on a separate conveyor belt to the right of the user (see Figure 4.5). If the fish is correctly sorted out, e.g., it is wounded, a green fish icon will pop up to indicate this. If the fish sorted out is not wounded, a red fish pops up to tell the user that it was mistakenly sorted out.

Packing

At the packing station, the goal is to inspect the fish packed in boxes that arrive at the conveyor belt. If they are not neatly placed into the box, this will have to be corrected. In Figure 4.6a fish arriving in boxes at the conveyor belt can be seen. While the box is transported along the belt, the user needs to pick up the misplaced fish and place it nicely back as seen in Figure 4.6b.



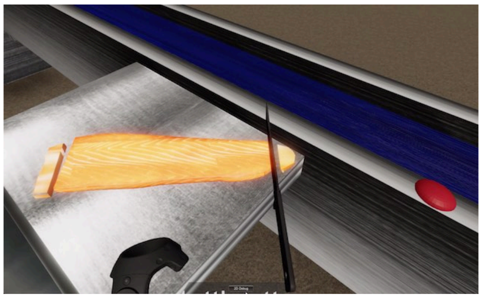
(a) Misplaced fish arriving at the conveyor belt. The user has to pick up the fish and place it nicely back into the box.

(b) The fish is placed nicely back into the box and disappears along the conveyor belt.

Figure 4.6: The packing stations, where the user needs to place misplaced fish nicely into their boxes.

Filet Cutting

The last station at the processing facility is filet cutting, and it is reached by following the black markers on the floor, through a door (see Figure 4.4b). The goal of the task at this station is to identify and remove excess fat from filets arriving at a conveyor belt. When the fat has been removed from the filet, it can be sent on its way down the conveyor belt. In Figure 4.7a the user is cutting of pieces of excess fat, while in Figure 4.7b a filet with no excess fat is depicted, and no further action is required.



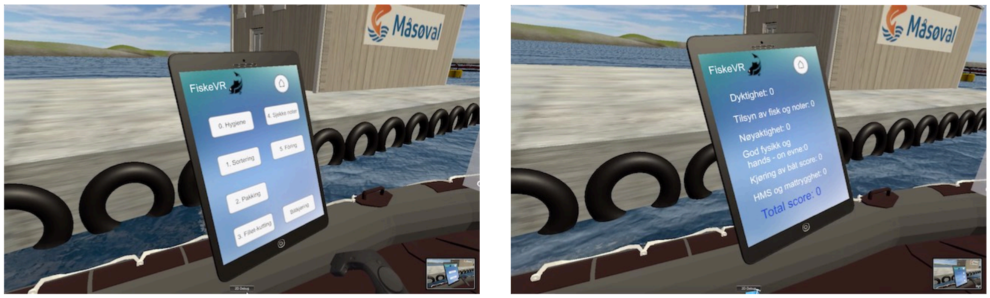
(a) The user removing left over fat from a filet.

(b) A filet that has no left over fat, so it does not need trimming.

Figure 4.7: The filet cutting station, where the user has to identify filets with left over fat and remove it.

4.1.5 Tablet

The user has an in-game tablet at her left hip which can be used to inspect tasks and points. To pick up the tablet, the user will need to reach down and grab it with one of the controllers, then use the other controller to navigate the menus using the Hand Pattern. All tasks available can be inspected at the tablet (see Figure 4.8a), so the user knows what is possible to do in the game. Tasks will have a tick-mark when completed, so the user knows what is done and what needs to be done. Tasks the users do when playing the virtual internship will earn them points (Figure 4.8b), which can be inspected with the tablet, where a break down of points can be viewed.



(a) The menu showing the different tasks in the game.

(b) The menu showing the break down of points earned during gameplay.

Figure 4.8: The in-game tablet the user can use to keep track of tasks and points.

4.2 Results

This section presents the collected data and analysis of them for the FiskeVR application. The data collection, selection, and analysis methods can be read about in Chapter 3.

4.2.1 Forskningsdagene and Researchers Night

Two events for young students were attended, where they could test the application and answer a volunteer questionnaire after testing. People in the age 14-18 attend both events, and everyone was free to try out the application at the IMTEL stand. The total number of respondents for these events is 23, with 21 male and 2 female with a median age of 16 years. The questionnaire used in these tests were short and brief with each statement rated by the Likert-scale. The statements can be seen in Table 4.1.

Figure 4.9 shows the results of the questionnaire. Most respondents agreed to some degree with the statements with the average score for each statement being: S1 (4.52), S2 (4.09), S3 (4.35). S2 received the lowest average score,

ID	Statement
S1	I enjoyed using the FiskeVR application
S2	After completion I had a better understanding of what the job depicted is.
S3	Such applications should be used as career guidance at schools.

Table 4.1: The statements related to the FiskeVR application, NAV Demo 10 of September 2018.

even though it received a relatively high score. One of the main points in the game is that the users should learn something about the workplace, and what they do there. After use, the users should walk away with a little more knowledge about the simulated workplace. The respondents are positive in using this type of virtual internship in career guidance at schools, and they seem to feel it can provide value. Additionally, there was a chance for users to comment on what could be improved in the application, answered received were:

- Higher frames per second (FPS)
- Better graphics
- In-game tutorial

Higher FPS and better graphics can be two features that are difficult to achieve simultaneously, as more advanced graphics have a chance to lower the FPS, and higher FPS would then possibly require lower graphics and less visual effects. An alternative is to optimize 3D models, textures, and rendering effects to be able to deliver a satisfying graphical experience at a decent FPS. Another factor is the hardware that is used to run the application, which can be improved to increase rendering capabilities. In-game tutorial is a point that can reflect the lack of visual or auditory guidance in the game.

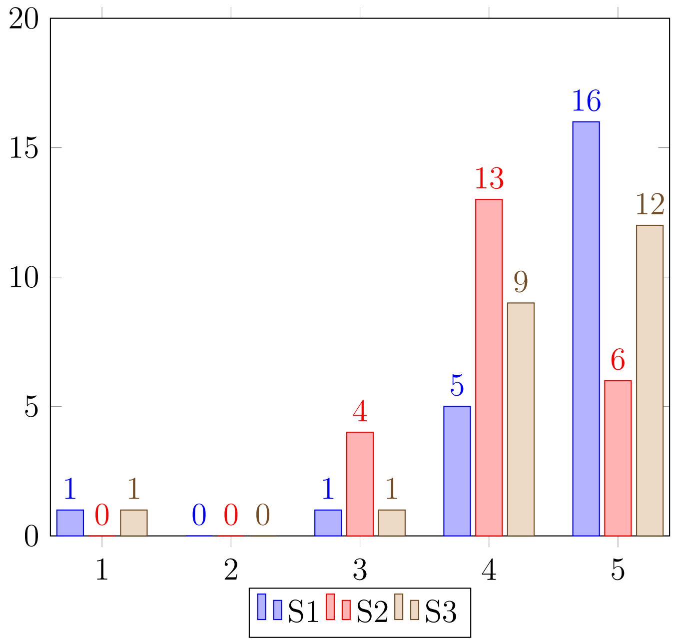


Figure 4.9: The answers collected at stands with FiskeVR. **S1**: "I enjoyed using the FiskeVR application" **S2**: "After completion i had a better understanding of what the job depicted is" **S3**: "Such applications should be used as career guidance at schools"

4.2.2 NAV-test: 10th of September

At 10 of September 2018, a test session with NAV users and employees was held at the IMTEL VR Lab at Dragvoll NTNU. In total, there were six job seekers and one employee attending the demo. One of the job seekers is removed from the results here, as the person only answered 3 of 15 questions for the FiskeVR application. The median age of the respondents is 24. Three facilitators were testing the FiskeVR application, where the author was one of them. In Table 4.2 the average results of statements rated with the Likert Rating-scale can be seen. The *visual quality* received an average score of 3.86, and the quality of the interactions in the application received an average of 3.57. Many young people today are used to computer games with high-quality graphics and well-implemented interactions, so it is important for a serious game like FiskeVR to go as far as possible in these areas, to meet the youth's expectations. A serious game should supply all components of a regular game; then the pedagogy should come as a subordinate, but nevertheless an important element.

ID	Statement	Average Score
2a1	How would you describe the visual quality of the application?	3.86
2a3	How would you describe the quality of the interactions in the application?	3.57

Table 4.2: The statements related to the FiskeVR application, NAV Demo 10 of September 2018.

In Table 4.3 average results of statements rated with the Agreement-scale can be seen. In general, the application seems to make users feel they get useful information about the task and workplace, and they enjoy using it. **2a9**: "The tasks I were given felt realistic", **2a12**: "After completion I had a better understanding of what the job depicted is" and **2a13**: "This application gave me insight into how I should complete certain tasks at the workplace" all received an average score of over 4.0. This points to the fact that the users feel it is a realistic depiction of a workplace, and that they both learn about the workplace itself and tasks that follows in such a job. **2a8**: "I felt seasick" was rated low (1.40) which is good. **2a11**: "The application gave me motivation to apply for a similar job" received a fairly low score (2.83). This, of course, will be highly individual and depends on the person and their preferences.

Table 4.3: The statements related to the FiskeVR application using the Agreement-scale. NAV Test 10 of September 2018.

ID	Question	Average Score
2a4	I enjoyed using the application	4.00

Continuation of Table 4.3		
ID	**Question**	**Average Score**
2a6	The tasks that where given were easy to complete	3.67
2a7	There was a lot in the application that felt illogical or unpredictable.	2.17
2a8	I felt sea-sick	1.40
2a9	The tasks I were given felt realistic	4.50
2a10	I am interested in this type of job	2.67
2a11	The application gave me motivation to apply for a similar job	2.83
2a12	After completion I had a better understanding of what the job depicted is	4.33
2a13	This application gave me insight into how I should complete certain tasks at the workplace	4.14
2a17	Timelimits on tasks would have made the application more interesting	3.17
2a18	Cooperation would have made the application more interesting	4.17
2a19	Access to other people's scores would have made the application more interesting (leaderboard)	3.00
2a20	An exciting story would have made the application more interesting	2.80

As of incorporating new features, the most well-received feature was **2a18** about bringing cooperation into the application. This may be a natural feature to incorporate into a virtual internship, as most jobs include some degree of cooperation and communication between coworkers. The lowest scoring feature suggestion was **2a20**: "An exciting story". It could be that the users feel an exciting story is not needed in order to represent a workplace. The users seem to be unsure on whether adding a leader board or similar would have a good impact on the game, as **2a19**: "Access to other people's score would have made the application more interesting" received an average of 3.00. In the textual answers, the main points were better graphics and better auditory quality which seems to be in line with previous results.

4.2.3 NAV Evaluation: 30th of October

On the 30th of October 2018, a larger evaluation test with both employees and job seekers was held at the IMTEL VR Lab at NTNU Dragvoll. In total, 27 people were attending the tests. Two respondents are removed from these results, as they did not identify themselves as employees or job seekers and answered both the employee and job seeker part of the questionnaire. After exclusion of these two, there were 25 total respondents. Of these, there were 19 job seekers and 6 employees. The respondents had a median age of 23.5, which is similar to the previous demo. There were three facilitators for testing the FiskeVR application, where the author was one of them. The results from the

ID	Statement	Average Score	Median Score
2a1	How would you describe the visual quality of the application?	3.70	4
2a2	How would you describe the auditory quality of the application (sounds, dialogue)?	3.45	3.5
2a3	How would you describe the quality of the interactions in the application?	3.77	4

Table 4.4: Statements the respondents rated by the Likert Rating-scale. NAV Test, 30th October 2018.

statements rated by the Rating-scale can be seen in Table 4.4. The results with this bigger group of respondents for statement 2a1 and 2a3 is similar to the result from the smaller demo 4.2.2 with averages of 3.70 and 3.77 respectively. The textual answers revealed that the graphical quality of the application should be more realistic. The questionnaire given to this bigger group differs slightly from the previous in the smaller demo and includes a statement about the auditory quality of the game. The auditory quality was rated slightly lower than the other statements. In the textual section, some comments requested more sound from the environment and production machines. These results point to that the game does not live up to the youth's expectations when it comes to graphics and auditory quality.

Table 4.5: Statements the respondents rated by the Likert Agreement-scale. NAV Test, 30th October 2018.

ID	**Statement**	**Average Score**	**Median Score**
2a4	I enjoyed using the application	4.13	4
2a6	The tasks that where given were easy to complete	4	4
2a7	There was a lot in the application that felt illogical or unpredictable.	2.04	2
2a8	I felt sea-sick	2.27	1
2a9	The tasks I were given felt realistic	4.19	4
2a12	After completion I had a better understanding of what the job depicted is	4.09	4
2a17	Timelimits on tasks would have made the application more interesting	3.24	3
2a18	Cooperation would have made the application more interesting	3.87	4

The results from the second part of the questionnaire can be seen in Table 4.5. As in the smaller demo, **2a9**: “The tasks I were given felt realistic” and **2a12**: “After completion I had a better understanding of what the job depicted is” received over 4.0 on average. The users seem to feel that the workplace presented is fairly realistic and that it provides them with good insight and information about the job. **2a7**: “There was a lot in the application that felt illogical or unpredictable” received an average of 2.04 which is fairly low, but maybe not as low as it should. Some comments that can be related to this statement given in the textual answers reveal some points:

1. The knife, when lost on the floor, can disappear and it is impossible to complete the task.

2. Driving the boat can be hard. The steering and throttle are very sensitive.

3. I needed explanation from the supervisor to understand everything.

The author notes that a great deal of explanation is needed in general for the users to understand and complete the tasks in the application. Since the game has no strict flow, the users can do what they want when they want, but many people are unsure of what they actually *can* do. This often results in the need for an explanation of what is possible or what they should do next. The information in the application is generally large amounts of text that the user needs to read through, which by experience many have no patience for.

Again, when the users are questioned about new features, cooperation seems to be a fairly popular feature, though it received a slightly lower score than on the smaller test. In addition to the questions about the application that were given to all participants, there were two parts where one was answered by the job seekers and one by the employees.

Jobseeker-specific Part This part of the questionnaire was only to be answered by job seekers or students, and the results are shown in Table 4.6. In total, there were 16 respondents in this part of the questionnaire, as three out of the original 19 job seekers did not answer it.

Table 4.6: Statements the job seekers rated by the Likert Agreement-scale. NAV Test, 30th October 2018.

ID	**Statement**	**Average Score**	**Median Score**
3.1	I want to use applications like this frequently through NAV/school	4.31	4.5
3.2	This type of application can make me more confident when applying for a job	4.00	4
3.3	This type of application can give me a better understanding of a job	4.38	4
3.4	This type of application can give me increased confidence when executing tasks at a workplace	3.88	4
3.6	This type of application should be a part of NAV's offer to job seekers	4.44	5
3.7	This type of application should be a part of career guidance at schools	4.67	5
3.9	The applications was easy to use	4.19	4
3.10	I would need assistance from a technical knowledgeable person to start using this type of application	2.94	3
3.11	I think the different parts of the applications fit well with each other	4.06	4
3.12	This type of application can give me motivation to apply for jobs	4.13	4

Continuation of Table 4.6			
ID	**Statement**	**Average Score**	**Median Score**
3.13	I would feel safer in a new job if I had tried this type of application beforehand	4	4
3.14	This type of application give me the feeling of presence at a workplace	3.50	4
I01	I think it was easy to feel immersed in FiskeVR on the HTC Vive	4.38	4

Statements **3.1**, **3.6** and **3.7**, which all is related to the usage of this type of application as career guidance at NAV and in schools, are well received with average scores well above 4.0. **3.7**: "This type of application should be a part of career guidance at schools" is the highest rated in this part of the questionnaire with an average score of 4.67. Additionally, **3.3**: "This type of application can give me a better understanding of a job" received an average score of 4.38. With all these statements receiving high scores, it can seem users feel the application is appropriate to give a good introduction to, and information about, a workplace, and to give users a sense of whether they would like this type of work or not.

On the other hand, **3.14** "This type of application give me the feeling of presence at a workplace" received a slightly lower score with an average of 3.50. As noted before, users have rated both the visual and auditory quality of the FiskeVR application somewhat low and in Section 2.1.2 we saw that immersion inducing techniques like graphical quality and auditory cues are important to enable the user to feel presence. When it comes to motivation and confidence, **3.2**: "This type of application can make me more confident when applying for a job", **3.12**: "This type of application can give me motivation to apply for jobs" and **3.13**: "I would feel safer in a new job if I had tried this type of application beforehand" received average scores of 4.0, 4.13 and 4.0 respectively. That is a positive outcome of the application in addition to the practical and factual information learned. In the textual answers for this part, relating to what could have made the concept better, an interesting point was "side quests or distractions can make it more realistic". This can be related to introducing multiple level goals or meta-goals as described in Section 2.5.2, where more skilled players can have additional things to do or goals of completing tasks at a certain speed or with a certain score.

Employee-specific Part This part of the questionnaire was only to be answered by NAV employees. In this situation, one person was a teacher at a school. In that case, the respondent answered the questions regarding *users* mentioned in the statements as *students* instead. The results of the questionnaire can be seen in Table 4.7. In total there were five respondents to this part, as one out of the original six attendees did not answer this part. In general, all statements regarding the use of the applications in career guidance at schools or NAV is very highly rated, with the employees seemingly being even more positive about its usage than the job seekers themselves with textual comments like "This is the future!". The employees also believe that the application will help motivate and inform the job seekers to apply for jobs.

Table 4.7: Statements the NAV employees rated by the Likert Agreement-scale. NAV Test, 30th October 2018.

ID	**Statement**	**Average Score**	**Median Score**
4.1	I want to use applications like this in my work with users	4.80	5
4.2	This type of application can help with preparing users for interviews	5	5

Continuation of Table 4.7			
ID	**Statement**	**Average Score**	**Median Score**
4.3	This type of application can introduce users for workplaces and typical workplace-related tasks	5	5
4.4	This type of application can give increased motivation in users for applying for jobs	4.60	5
4.5	This type of application can give users increased sense of achievement in the job seeking process	4.20	4
4.6	This type of application should be a part of NAV's service to job seekers	4.80	5
4.7	This type of application should be a part of career guidance at schools	4.80	5
4.8	A larger focus on game elements woulda have made the applications more interesting	4.00	4
4.9	The applications was easy to use	4.00	4
4.10	I would need assistance from a technical knowledgeable person to start using this type of application as a part of the NAV service	3.50	4
4.11	It would be easy to integrate this type of application as a part of the NAV service at my workplace	3.75	3.5

Focus Group Interviews

In addition to the questionnaires given to the users, they had the opportunity to volunteer to attend focus group interviews. This was done to provide more qualitative data and to investigate further what the users thought about the game.

In general, the users seem to feel the application is user-friendly. One point that came up was that the users felt the application needs an in-game, visual, or auditory way of introducing facts and tasks. They pointed out that it could be confusing and tiresome to both play the application, and continuously have to listen to the supervisor standing next to them. This is a point that comes up several times; the game needs more direction and tutorials for the players. They additionally noted that the boat controls were hard and unpredictable.

When it comes to the visual quality of the game, the users felt that the quality needs to get better. Along with better visual quality, it was outed that more visual fidelity was wanted. The users mentioned things like interactive objects, animals, or just even more environmental sounds. Even though the visual quality was not at a level they were satisfied with they admitted that the simulation felt quite realistic.

The users mentioned the importance of feeling presence, and that this was a big reason to why they enjoyed the game. They continued to say that it was great to be able to immerse themselves in a workplace and a job, to see how things are done at the workplace, being able to interact with objects and try out different tasks. “You get to actually experience how it is to work there, instead of just reading it from a book or watching some video.”

The users also mentioned an interesting point, that a danger with these simulations is that they are too “funny”, as in funnier than the actual job, possibly giving them higher expectations of the real-life workplace than what is the reality. They continued by saying that it is hard to accurately depict, for example, crisis, stressing situations, and physical exhaustion. Games generally focus on being enjoyable and fun, but maybe virtual internships need to be somewhat constrained in order not to depict the wrong picture?

The value the users seem to see in using these applications is good and interesting information and a sense of some practical experience of a job. They feel that it may be especially useful for people unsure about what career to choose, but then there would be a need for a lot more simulations than a single workplace. “It is a good opportunity to see how different workplaces are”. Some users admit that they are more interested in the workplace after playing.

Some reminiscent back to times when they were allowed to try out different jobs practically, through school programs. They then mention that they were usually set to the “boring” parts of a job and that they felt they did not get to experience the actual job and were demotivated to explore a career within that line of work further. Another point is that in general, it seems like people that are interested in technology, maybe even interested in VR, feel such applications have more potential than the less interested people. It could be that the users that are entirely new to VR will get a better experience as the uncertainty of using the technology fades and that the feeling of presence will increase as the technology feels more natural over time.

4.2.4 NAV Test: 26th of November

At the 26th of November, 2018 a test with employees was held at the IMTEL VR Lab at NTNU Dragvoll. In total, there were 13 respondents to the questionnaire, all employees. The median age of the respondents that gave their age was 52, but surprisingly, 6 out of 13 chose to not respond with their age. Nevertheless, the respondents in this demo were significantly older than in the previous demos. There were 3 facilitators during testing of the FiskeVR application, where the author was one of them.

The result of the questions, rated with the Likert Rating-scale is presented in

ID	Statement	Average Score
2a1	How would you describe the visual quality of the application?	4.23
2a2	How would you describe the auditory quality of the application?	3.75
2a3	How would you describe the quality of the interactions in the application?	3.85

Table 4.8: Statements rated by the Likert Rating-scale from the test with NAV employees at the 26th of November, 2018.

Table 4.8. The visual quality of the application received a noticeably higher rating than previously, with an average of 4.23 compared to 3.86 and 3.70 for the other demos, respectively. The biggest difference in the group of respondents for this demo is the fact that the respondents where on average much older, and solely employees, compared to a mixed group of employees and job seekers with a bigger group of young people for the other demos. It could be that the young people's expectations are higher due to a higher exposure of well-made, high-budget video games. Since the young job seekers or students are the actual target group, their needs should be prioritized in development. **2a2** *auditory quality* and **2a3** *interaction quality* did not have significantly changes in average score compared to the other demos, though slightly higher than earlier. The auditory quality was commented on with "I want better sound and more people in the room". It is a recurring theme that people feel the game is a bit *empty*, with few auditory cues, visual cues, and no animated NPCs[1] to interact with.

In Table 4.9 questions rated with the Agreement-scale is presented. There are no big changes in respect to the other demos here but some variances. **2a7**:

[1]NPC: non-player character, an computer controlled entity in the game.

ID	Statement	Average Score
2a4	I enjoyed using the application	4.15
2a6	The tasks that where given were easy to complete	3.77
2a7	There was a lot in the application that felt illogical or unpredictable.	2.46
2a8	I felt sea-sick	2.15
2a9	The tasks I were given felt realistic	3.92
2a12	After completion I had a better understanding of what the job depicted is	4.23
2a17	Timelimits on tasks would have made the application more interesting	3.08
2a18	Cooperation would have made the application more interesting	3.25

Table 4.9: Statements the respondents rated by the Likert Agreement-scale. NAV Test, 26th November 2018.

"There was a lot in the application that felt illogical or unpredictable" was in a higher degree agreed to by this group of testers in comparison to the other demos. Some comments that relate to this is:

- The knife disappeared
- The knife is hard to use
- Hard to put fish in the boxes

These are irregularities that should be eliminated in the long run to not demotivate the users. The slightly lower score on **2a9**"The tasks I were given felt realistic" can also be connected to the above comments. If the game behaves in entirely unexpected ways, it can break the feeling of realism. **2a12**: "After completion I had a better understanding of what the job depicted is" gained an average of 4.23, reinforcing the fact that the application provides a good way for people to learn about a workplace. There is no clear indication if features like cooperation or time limits are wanted. This may call for experimentation to decide if they have any utility.

4.3 Discussion

This section will discuss results from all the tests executed in the case study of the FiskeVR application.

4.3.1 Virtual internships as Serious Games

The goal of a virtual internship is first, and foremost, to give the users an impression of how a workplace is to work at, and relevant information and facts about the workplace. A virtual internship is a *serious game* as described in Section 2.5.1, and it is important to identify which learning outcomes are wanted, and keep them in mind under design and development. The main relevant learning outcome of these virtual internships is affective, influencing the user's emotions and attitude towards a job. Plus, they will provide both declarative knowledge about the workplace and the tasks, and to some degree, procedural knowledge about the tasks. The last two learning outcomes are mainly tools towards inducing the aimed for affective learning outcome. Of course, additional unintended learning outcomes might exist. Furthermore, dependent on the workplace depicted, learning outcomes like communication or decision making can be a result of some of the tasks the user has to do.

It seems that users, in general, feel that the application gives them relevant information and a sense of how the fishing industry is as a workplace, with high scores for realism, understanding of the workplace, and insight into how specific tasks are performed at the workplace. The applications seem to have potential as a tool for career guidance in both schools and NAV, with the most popular option being schools (from the job seekers point of view). The fact that they think it is an excellent tool for informing about a career reinforces that the real value from usage is that they learn something and gain insight.

NAV employees are equally, if not more, enthusiastic about the usage of the application as a tool to guide and motivate the unemployed. Many participants in the tests request more workplaces to be available. To make this possible further experimentation with the development process of such applications should be done and more applications should be developed.

4.3.2 Utilizing Game Design Theory

It is essential to look at a virtual internship as a game, while still making sure that the amount of resources and time in making it to much of a game is held in check. Using techniques from game design theory can, however, help these applications be more exciting and attractive for young people to use.

Challenge

The users do enjoy the tasks, but they offer little ways of growing skills. This is, however, something that can be regarded as acceptable, as the learning outcomes of these applications are not about growing the users skill in a profession. This is a returning factor through the literature study and this case study and seems like a point where a regular game and a virtual internship does go separate ways. This is not to say that future applications should not experiment with implementing more ways of adjusting difficulty and focus more on the fact that the user should grow their skill-set, but it is not necessarily a pressing area to investigate any further. The way NAV wants to use these applications have to be regarded; they are not *games* in the traditional sense, they are informational virtual representations of workplaces, with examples of tasks that workers encounter there. They are tools that can be used to inform young people, but it does not mean abandoning traditional tools like visits to companies, and real-life internships. This separation and reduction in the scope of virtual internships are important as any expectations for these application to be fully fledged games with truly mesmerizing worlds and engaging tasks, offering a variable difficulty and vast amounts of content are simply not viable. There is no evidence that these applications cannot be all that, but it might not be worthwhile to make them so, as development is a balance between cost and benefit.

In general, it is necessary to supply the users with clear goals, feedback on progress towards said goals, and information on completion of the task or goal. Ideally, the application is presenting enough information along the way so that users will be able to complete the tasks without interference by a supervisor or another external person.

Fantasy

The fantasy in a virtual internship is naturally intrinsic, as an extrinsic fantasy would make little sense, and possibly be detrimental to the experience. The specific implementation of the fantasy will vary from application to application, where each new application will require new assets, new metaphors and a new *world* to create for the designers and developers of the application. The creation of the world and metaphors is where it seems like the bulk of the work rests in a new application; representing a completely new workplace realistically, gathering data and information and turning this into a digital environment that the user can navigate and interact. In general, users seem pleased with the realism of the current application, and it is important that the fantasy is close enough to reality that the users get a realistic picture of the job. It is also equally important that the industry feels like the representation is realistic and that it supplies important information. Balancing the line between realism and virtual metaphors in a way that does not make the experience unbearable and difficult to understand, but yet represents a realistic picture of the job is required. This can call for not overusing techniques that make games fun, as there is a danger where the digital representation is too fun and enjoyable, compared to the real-life job. This is also commented on by a user of the application, where a concern is that they do not get a realistic picture, but rather a fun “gamified” version of the job. Still utilizing certain techniques to induce enjoyment and other emotions should be utilized thoughtfully to attract the user’s focus; this is when they learn and assimilate information best.

Curiosity

Curiosity is important for enabling the user to feel the motivation to learn and explore the application. A virtual internship should aim to induce both *sensory* and *cognitive* curiosity in order to motivate them to learn more about the workplace.

Sensory Curiosity Better graphics and audio is one of the most requested improvements of the application. Some users feel the current state of audio and graphics breaks the realism a bit, and that they would feel more presence at the workplace with higher fidelity both visually and auditory. As explained in Section 2.1.2, these elements are the base foundation of what is needed to create immersion, which again can induce presence. Young people do expect high quality from commercial entertainment games, and it is not far fetched to think that they will be disappointed if the quality of these virtual internships is prominently lower. However, it is also important to note that even though the graphics are not on the level with modern commercial games, the users do feel a sense of realism. There is an important factor in balancing the time and effort used to improve the quality versus the benefits gained, as earlier mentioned. Better graphics and audio will require more development effort, and will also most likely demand more of the hardware the applications run on, making them both more expensive to develop and more expensive to use.

Cognitive Curiosity It can be tough to ensure the activation of a user's cognitive curiosity, as the user is such a big element in the process. As described in Section 2.5.3, people want to learn new things in areas where they already have some knowledge. When they know enough to know there is a hole in their knowledge, they want to cover it. It is hard to assess if FiskeVR induces this effect, but most likely it does, however, in a varying degree depending on the user. Steps that might be beneficial in inducing this is giving the user *some* information, but not all, urging them to further explore the world in order to understand and complete tasks.

4.3.3 Tutorials and information

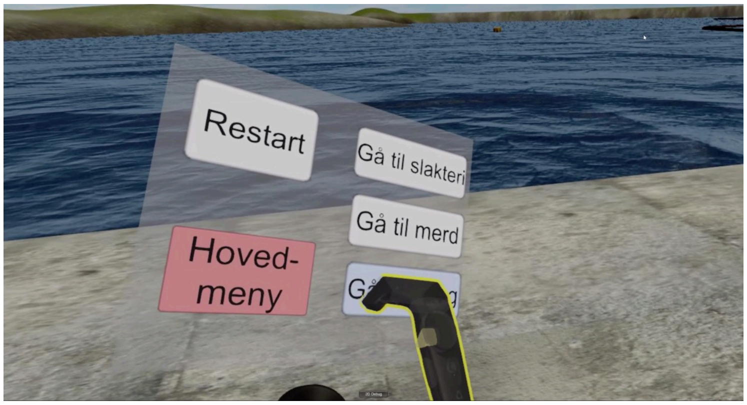
The application does provide both 360 videos and textual information that explains the context and process of each task. The videos are, however, not very instructional, but do give great insight into how it looks in the processing facility. The instructions at each task heavily rely on large amounts of text that the user has to read, which is not always the best option. It should be experimented with spreading information more around the world, making it available to the user when it is needed. This is one of the strengths of games found in Section 2.5. From results, and observations during the tests, the application sometimes require an excessive amount explanation from the supervisor as the user will be confused in what to do next and how to do it. The fact that VR technology is so unfamiliar to many users does seem to be a factor in this regard. Users do comment that an in-game tutorial would be appreciated, that could show how to do things.

4.3.4 Feedback, tasks, and progress

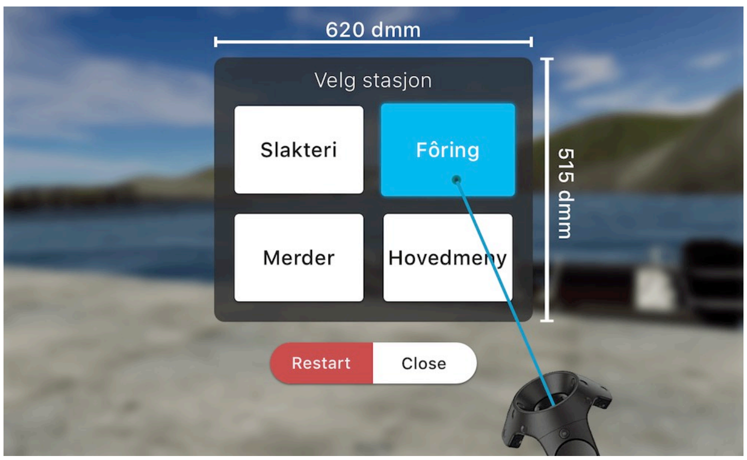
The score system in the application is generally a useful feature, as a fundamental aspect of a game with tasks is offering a way for the player to check their progress and how they are performing. The feedback on whether they are doing right or wrong is a little varying throughout the application. The sorting station has the most explicit feedback, with either a red or green fish popping up to show if the action was correct or incorrect. The feeding station has the least amount of feedback, which in reality, is none as it is hard to tell when the feeding is complete. The tablet that tracks completed tasks and scores in-game does make a sound whenever the player completes a task, but there is no introduction in-game to what this sound means or that it is, in fact, the tablet that is emitting it. It is vital with clear goals but also proper feedback on the progress and current score in regards to these goals.

4.3.5 User Interface

The user interface in the current application is fair, and most users seem to grasp it after some explanation. The pattern for interaction used is the *Hand Pattern* (Section 2.2.2), which some users are confused by in the beginning. In general, more consistent design to where the menus and information is located could have a positive effect on both frequency of usage and how understandable it is to the user. As of now, the user have to check the tablet to find a specific task, but have to open another menu to navigate there. Also, the Point Pattern can be a more intuitive selection pattern for menus. Here it is important that menu items and buttons are large enough, and by specifying the design in dmm's (Section 2.4.1) this can be ensured, for the intended viewing distance of the interface element (or *screen*). Developing a more rigid way to specify, design, and implement UI elements across all virtual internships could bring more continuity to each application, but also across the concept. As an example of how to specify a menu, the in-game menu is chosen as a study case. The original menu can be seen in Figure 4.10a. Here the menu is locked to the users' left-hand, while they will have to use the right-hand to select the different buttons. This menu uses the Hand Pattern, where the user takes the controller physically "inside" the button and pulls the controller trigger. If the user keeps their left-hand down by their hip when bringing up the menu, they might not even see the menu. In Figure 4.10b a specification for an alternative menu, using the Point Pattern is presented. Here the user will have a pointer originating from the controller, and by pointing at a button, the button will enlarge and change color to indicate that it is currently the selected one. The user will then be able to pull the controller trigger to *select* the button.



(a) The current in-game menu.



(b) An alternative menu design, using the Point Pattern for selection. The menu is placed in the world space at a certain distance from the user. Here specified in dmm to ensure readability at the intended viewing distance.

Figure 4.10: The current in-game menu design, and the proposed redesign.

Chapter 5

Design And Creation of WindturbineVR

This chapter will introduce the application, and relevant tools. The process of design and creation can be seen in Section 3.1.2. In the following chapters, the design and creation of a new virtual internship, where the user will explore how it is to be a wind turbine technician, will be presented. Each of the three main iterations are presented in separate chapters: First Iteration (Chapter 6), Second Iteration (Chapter 7), and Third Iteration (Chapter 8). The final product is presented in Section 8.7, and results from evaluation tests for the final product is presented in 8.8.

5.1 Application - WindturbineVR

This section gives an introduction to the developed application.

5.1.1 How does a wind turbine work?

Before starting the design and development, a crucial element is to understand how a wind turbine works, and what tasks are relevant for a wind turbine technician. A wind turbine does, in general, consist of three main parts: the tower, the nacelle and the rotor with blades. As seen in Figure 5.1, when the wind hits the blades, the rotor will turn and drive an electric generator generating electricity. This electricity is sent down the tower to a transformer, transforming it and sending it out on the grid (Good Energy 2018). There are both electrical and mechanical systems in the wind turbine that will need maintenance by a wind turbine technician. As a wind turbine technician, it is required to have education or experience in electrical systems and/or mechanical systems, and general tasks in a workday are; debugging, maintenance and preventive work on turbines, transformers, and generators, and be able to work in great heights while doing so (TrønderEnergi n.d.).

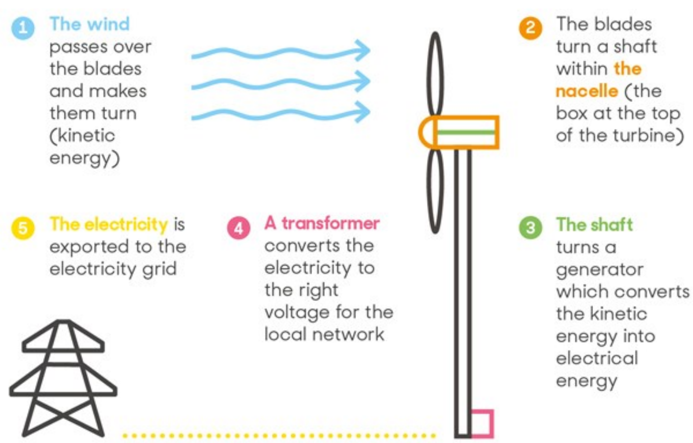


Figure 5.1: Schematic displaying how a wind turbine converts wind power to electric energy (Good Energy 2018).

5.1.2 Learning Outcomes

As found in the literature study, learning outcomes are good to have in mind throughout the development of the application, or *serious game*. The main outcome of a virtual internship is affective, where the internal state of the user can be affected. Additionally, declarative knowledge about the workplace and tasks is important, and in truth needed to be able to have the wanted outcome.

They are meant to help the users take more informed decisions about their career.

5.1.3 Main parts

Table 5.1 presents the three main parts of the application. The developed MVP consists of the internship and the main menu. The interactive tutorial has been partly developed, but can and should be further extended. All these parts are fully presented in Section 8.7, where the complete implementation of teh application is described.

ID	Description	MVP	Developed
C1	Main Menu	Yes	Yes
C2	Interactive Tutorial	No	Partly
C3	Wind Turbine Technician Internship	Yes	Yes

Table 5.1: Three main parts of WindturbineVR.

Main Menu

The main menu is where the user starts their journey and choose what part of the application to enter. The main menu is a 3D environment, with clear, concise menu items for the player to choose. Interacting with the menu will use the Point Pattern described in Section 2.2.2.

Interactive Tutorial

The interactive tutorial introduces the player to different interaction and movement patterns used in the game. This is supposed to be a more general introduction to the technology for the unfamiliar, to make the gameplay more smooth and easier to learn. This is a part that should be further developed and reused for the entire virtual internship project as a whole. This project introduces only the teleport functionality through the dedicated interactive tutorial.

Wind Turbine Technician Internship

The tasks in a wind turbine are many, and fully implementing functionality for them would be impossible in the time frame and with the resources available for this thesis. TrønderEnergi supplied a list of relevant tasks, where tasks that they could not disclose because of liability reasons is removed. Industry secrets can be a challenge in some industries, where secrets are guarded fiercely, and where the tasks the workers execute and details about them would not be possible to gain specific information on because of industry secrets and contracts between suppliers. The tasks TrønderEnergi could supply is listed in Table 5.2. These tasks acted as inspiration for creating relevant tasks in the turbine, to represent the line of work and still being “simple” enough for VR applications, and achievable with the time and resources available. Section 8.7 includes a presentation of the tasks. Possible further development is implementing additional tasks.

Table 5.2: A list of tasks received from TrønderEnergi that they deemed relevant.

ID	**Task**
T1	Replace the fire extinguisher with one who have been controlled and have a ok inspection date.
T2.1	Check the inspection date of the first aid kit and verify that it is undamaged.
T2.2	Verify that the inner plastic film of the first aid kit is undamaged.
T2.3	Replace the first aid kit if either T2.1 or T2.2 is not in order.
T3.1	Perform a visual and functional check of the entire rotor lock system.
T3.2	Completely set the rotor lock.
T3.3	Check oil level, and refill with Renolin PG46 if not ok.

Continuation of Table 5.2	
ID	**Task**
T4.1	Perform a visual inspection of the hatch to the nacelle basement.
T4.2	Perform a visual inspection of the winch hatch.
T5	Check cables between control card rectifiers and thyristors. Replace if damaged.
T6	Check cabinet fans in: excitation controller box, rectifier cabinet, nacelle control cabinet and generator filter cabinets.
T7	Check the central lubrication system and replenish grease reservoir if necessary.
T8.1	If necessary update the beacon system software.
T8.2	Visually check the beacon for damage, corrosion and dirt.

5.2 Tools

This section introduces tools, SDKs, and other software used during the development process. Knowing the tools is essential to understand some of the development in the following three chapters. The reasoning behind the selection of tools, where other tools are available and where the alternatives are deemed valid, will also be presented.

5.2.1 Unity

The game engine used in WindturbineVR is Unity, which is the same as for previous virtual internships. It is flexible and handles both 3D and 2D development, physics, graphical post-processing effects, and UI. Also, it is free and approachable, and therefore has a large online community. It has a content-rich asset store, where premade models, code, and systems can be downloaded or bought to speed up the development process. It uses either C# or Javascript as the programming language. It supports a wide range of platforms, including consoles, mobile, web, desktop, and XR. The Unity version used in this project is 2018.3, and relevant documentation used in the process is for this version of Unity. This project is written in C#.

The alternative: Unreal Engine

The only real alternative to Unity is Unreal Engine, which also features support for XR development. It is a powerful game engine, using visual scripting or C++ for programming. Just as Unity, impressive scenes and games can be made, and one can say that it in a huge degree comes down to personal preference. As earlier virtual internships are developed in Unity, it was deemed that changing the game engine would not be beneficial as eventual re-usability of assets and code would be hard. Also, the author has minimal experience in C++ programming, which could prove to be a barrier to development.

5.2.2 Blender

Blender is an open-source 3D creation suite (Blender Foundation 2019a). It is a cross-platform (Windows, Linux, and macOS) tool and supports both 3D modeling, animation, texturing, simulation, and rendering. By directly importing **.blend**-files into Unity, models can later be edited by double clicking on the model in Unity, which will open Blender. Changes done to the model in Blender will be immediately applied and imported in Unity on save. A drawback when directly importing models from Blender is that they will not correctly show up in the Unity project if Blender is not installed on the machine, but as Blender is free, open-source and relatively small, it is a price worth paying for the convenient and effective workflow this enables. No alternatives to Blender where considered.

5.2.3 Autodesk Recap

Autodesk Recap can visualize point clouds with color data in an easy-to-use interface. It can be used to combine multiple 3D scans into one point cloud that can be navigated and inspected, both in 3D and real view. This makes it possible to navigate and inspect 3D scans of the wind turbine supplied by TrønderEnergi. This software was mostly used as an aide when constructing the wind turbine and was vital in being able to reconstruct the turbine realistically.

5.2.4 Sketch

Sketch is a vector graphics and user interface design editor. It is useful to draw icons and 2D user interface elements that can be exported and used in Unity as UI elements. It has also been used to construct some of the figures in this thesis.

5.2.5 Virtual Reality Toolkit (VRTK)

The base of a VR project is the SDKs used for connecting with the VR hardware. Earlier virtual internships build upon Valve's SteamVR SDK for virtual reality functionality. To explore and experiment, for this application Virtual Reality Toolkit (VRTK) was used as the foundation. There are several SDKs available for use when developing VR applications in Unity; SteamVR, Windows MR, Unity XR, and OcculusVR. VRTK is a toolkit that abstracts away these SDKs into a unified API with an additional massive collection of utility scripts to speed up development. For this application, Virtual Reality Toolkit is chosen, as it provides a lot of utility and support for a wide variety of HMDs. The drawback of choosing something else than previous Virtual Internships is that functionality that may exist is considerably harder to reuse, and in many cases impossible. However, utilizing VRTK will also provide information on using new tools to increase efficiency when developing these types of applications, which is useful in the project. The complete documentation for VRTK resides at `https://vrtoolkit.readme.io/v3.3.0`

Since VRTK is essentially an abstraction layer for existing VR SDKs, major updates to the underlying software can be incompatible without patching the VRTK software. This will, however, change for the future version of VRTK, where it goes truly hardware agnostic. It will no longer depend on other SDKs implementation and utilizes Unity's built-in XR support (Heaney 2019).

5.2.6 Git

Git was used for source control in the project, with Git LFS enabled for large files. In Unity, all asset files were forced to be serialized to text, in order to better support Git.

5.3 Hardware

There are three main hardware systems the application is tested with, and will support. It is a possibility that it does support other hardware than described in this section, because of device support through SteamVR, or by adding additional SDKs supported by VRTK, but this is not explicitly tested.

5.3.1 HTC Vive and Vive Pro

The HTC Vive comes in two versions: the Original and Pro, where the Pro version has an increased resolution on the display of the HMD. The Vive Pro can be seen in Figure 5.2. The Vive systems use mounted base stations that bathe the play area in IR light, which the HMD and the hand controllers use to track their position. This means that it is not a very portable system, which is why, when testing the application somewhere else than the IMTEL VR Lab, a Windows Mixed Reality system (see Section 5.3.2) is usually used. However, the tracking accuracy and speed are noticeably better on a Vive system compared to less advanced systems like Windows MR. For future reference, as some chapters will reference the buttons of the controllers, Figure 5.3 depicts and describes all the different buttons. When mentioning these during the development description, this controller scheme is followed.

Figure 5.2: HTC Vive Pro HMD, controllers and base stations.

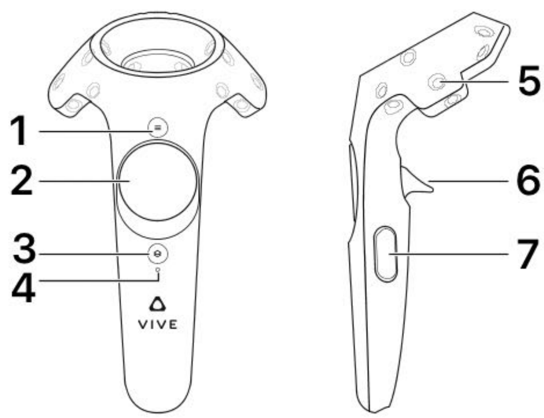


Figure 5.3: Buttons on a Vive controller; 1: Menu button, 2: Touchpad, 3: System button, 4: Light indicator, 5: Sensor points, 6: Trigger, 7: Grip button. Image adapted from `https://www.evetech.co.za/repository/ProductImages/htc-vive-controller-730px-v1.jpg`

5.3.2 Windows MR

Windows MR is Microsoft's VR brand, where several manufacturers like HP, Dell, Asus, and Samsung produce HMDs and controllers branded as Windows

MR systems. These systems are, in general, a lot more affordable than the Vive Pro and usually have a lower resolution display, smaller FOV, and many of them have no built-in headphones (you can connect your external headphones). Unlike the HTC Vive, Windows MR does not need base stations for tracking and is therefore very practical when portability is an important factor. Therefore, most tests not done at the IMTEL VR Lab is with a Windows MR system.

5.3.3 Oculus Rift

The Oculus Rift is the middle way between how Windows MR and HTC Vive does tracking. It does employ base stations like the Vive, but these can be placed on a desk facing towards the play area and are more flexible than the HTC Vive base stations. These were included as a supported system since TrønderEnergi had these at the VR room in their office.

Chapter 6

Development - First Iteration

The first iteration lasted from 15th of January to 14th of February. The focus of the development in this period was to construct basic internal models of a wind turbine, landscape and implement basic virtual reality functionality to move around the world and interact with objects. This section will describe the development during this iteration, and end with results from tests performed during the iteration.

6.1 Features

The first version was planned to include three main parts (see Section 5.1), regarded as the main parts of the application: the main menu, the virtual internship itself, and an interactive tutorial. The virtual internship includes an exterior scene with landscape and wind turbines and an interior scene of the turbine with electrical and mechanical components. The interactive tutorial developed contains a teleport tutorial. The application should support the whole range of the most common HMDs, including HTC Vive, Windows MR, and Oculus Rift.

The specific features, or implemented fixes (fixes are not applicable for this iteration as it is the first), is represented in Table 6.1, with an ID and de-

ID	**Description**	Relevant Sections
F1	Set up development environment.	6.2
F2	Basic Unity scene with VR support and possibility for locomotion and interactivity.	6.3
F3	Wind turbine tower interior with possibility to climb up to the top.	6.4
F4	An exterior landscape depicting Hitra	6.5
F5	Include an external model of a wind turbine that the player can stand on top of.	None
F6	A main menu where the user can choose between tutorial and starting the actual game.	6.6
F7	A teleport tutorial that introduces the concept of teleporting to the user.	6.7
F8	An in game menu where the user can quickly navigate to the different parts of the application.	6.8

Table 6.1: The features for the first iteration of development.

scription. The following sections in this chapter will describe implementation details with brief discussions where applicable. Not all features require a separate section to explain how and why, therefore, not all entries will lead to a section in the table.

6.2 Development environment

This section will briefly describe setting up the development environment.

6.2.1 Git

Setting up Git with Unity requires adjustments with both settings in Unity and Git to get everything to work well together. First, turn on *Visible Meta Files* and *asset text serialization* in the Unity editor. As described by Thomson (2018), it is then important to configure Git with the correct *gitignore*, *gitattributes* and configuration for Git Large File Storage (Git LFS). The gitignore will ensure noise, and temporary files from the editor are not checked into source control. The applied gitattributes will ensure proper storage with Git LFS and handling merging of advanced files like Unity's yaml files.

6.2.2 Virtual Reality SDKs

As described in Section 5.2.5, VRTK is used as the VR SDK for this project. Therefore, compatible versions of VRTK, and the underlying OEM SDKs for the supported systems in the project will need to be installed. Table 6.2 presents the versions of each SDK used in this project.

SDK	**Version**
VRTK	3.3
SteamVR	1.2.3
OculusVR (OVR)	1.28.0

Table 6.2: The SDKs and the versions used in the project.

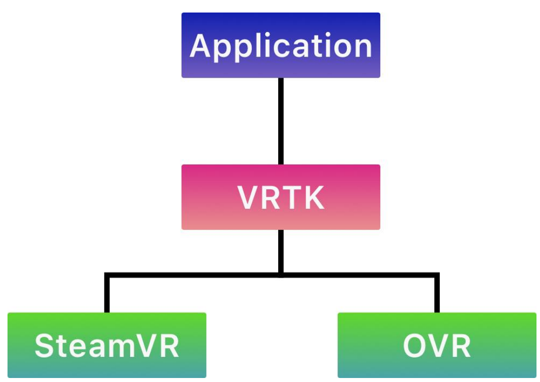


Figure 6.1: VRTK abstracts away the underlying VR SDKs and enables a unified API to develop VR applications.

When running the application, VRTK will automatically use the correct underlying SDK based on the connected VR HMD to the computer at the launch of the application. Connecting an HTC Vive will activate SteamVR while connecting an Oculus Rift will activate OVR. VRTK acts as a common abstraction for other SDKs, providing a single unified API to use. The arrangement of the SDKs in the project is presented in Figure 6.1.

6.3 Basic scene with VR support

The basic setup of a VR scene with VRTK contains three main components: an **SDK Manager**, **device aliases** and **locomotion scripts**. The SDK Manager ensures that the correct SDK is used, according to what HMD is connected to the computer. *Device aliases* are used as script targets for VR devices, like the HMD, controllers, etc. This way, scripts can be added to the controllers without knowing which SDK will be activated. The scripts added to the aliases are added to the corresponding devices in the underlying SDKs at runtime. The *locomotion scripts* determine what type of player locomotion to use, and VRTK supplies a set of pre-made scripts that can be used. Additionally, more than one locomotion script can be applied in a single scene.

Enabling teleportation

To enable teleportation, a few things need to be set up. Firstly, an object representing the ground is needed. This object can be a simple 3D model like a plane or a more detailed terrain object; as long as it has a collider component (Unity 2018b). Teleportation involves the user pressing a button to enable a pointer, let the user point at a location in the scene, then when the button is released, the user will teleport to that location. For selecting the desired teleportation location a *VRTK_Pointer* and a *VRTK_BezierPointerRenderer* will need to be attached to the controller script aliases (VRTK 2018g)(VRTK 2018a). These scripts will listen for controller events to enable the pointer on button press and render a curved pointer at a location on the ground collider. The cursor rendered at the collision position between the pointer and the ground will change depending on whether the location is a valid location for the user to teleport.

Additionally, a locomotion script that enables the teleport action itself is needed. The script used here is the *VRTK_HeightAdjustTeleport*, which updates the player location in 3D space (x, y, z) and can automatically snap the player to the nearest ground object (VRTK 2018d).

Enabling interaction with objects

Another main feature of VR is the ability to interact with objects in the scene. The user should be able to pick up objects, use buttons, and open doors. VRTK includes a set of utilities to implement proper interaction mechanisms. Any object that is to be interacted with needs the *VRTK_InteractableObject* script attached to it (VRTK 2018e). This script marks the object as interactable so that touching, using, and grabbing can be enabled.

Touching The base interaction is to be able to touch objects, meaning that the application can register when the player touches an interactable object with their controllers. In order to enable touch interaction with interactable objects, the controller script alias will need a *VRTK_InteractTouch* script attached to it. This script actives touch events on interactable objects.

Using Using an object can be an abstract term, but in this case, it means performing an action with an object, by pressing a button on the hand controller that touches it. An example is pushing an interactive button: the player would need to touch the button with the controller, then press a button on the controller (for example the trigger) to *use* the button. To enable this feature with VRTK, the script aliases for the controllers will need a *VRTK_InteractUse* script attached. Objects that should receive use events have to be marked as *usable* in the editor options in Unity.

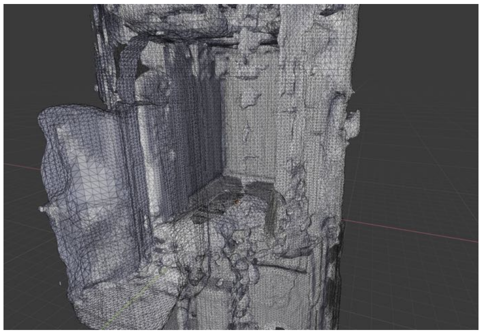
Grabbing Grabbing an object is technically just a specific *use* action, but requires some more scripts and setup. The controller script aliases will need a *VRTK_InteractGrab* script attached to them. Furthermore, the interactable object will need to have a grab mechanic script attached. The grab mechanic decides how the grab is performed, for example, grabbing an object can be done by adding the object as a child of the controller, it can be attached to the controller with a physics joint, or be forced to follow the transform of the controller. With these scripts set up, the player can pick up and place objects freely in the environment, as if in the real world.

Interaction Feedback Giving feedback on interactions in VR is important. One way to provide feedback on interactable objects is to highlight them. By attaching a *VRTK_InteractObjectHighlighter* script to an interactable object, the object can be highlighted with different colors depending on the state of it: used, grabbed or touched. On some objects, it can be useful to provide haptic feedback to the user; this can be done by using the *VRTK_InteractHaptics* script. The haptic feedback will only have an effect for VR systems with controllers that provide a possibility for haptic feedback; for example, the HTC Vive. Therefore, haptics should not be the only feedback provided, especially if the application supports systems that do not include haptic feedback in the controllers.

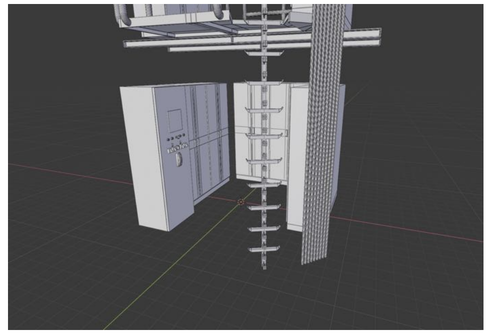
6.4 Creating the tower of the wind turbine

Any application like a virtual internship that effectively simulates a real-world workplace, needs a technique for bringing a real-life scene or component into the virtual world. The ideal scenario would be that all models were pre-existing and that no work in this area was needed, but this will seldom be the case. This area proved to be one of the most time-consuming tasks in recreating the workplace. Most companies producing wind turbines will most likely have extremely detailed 3D CAD models of the turbines they manufacture, but their willingness to give these out to the public would most likely be small. Additionally, extremely detailed models would incur a significant performance hit to the application. As a result, the internals of the tower in the wind turbine was manually created, then the model was imported into Unity.

From 3D scans to polygon meshes TrønderEnergi supplied 3D scans of one of their wind turbine in the form of point clouds and the resulting polygon meshes after an automatic software conversion. The ideal situation here would be to be able to use these generated mesh files directly in the application, but unfortunately, they are both incredible high fidelity (in terms of the number of vertices and polygons), and low in accuracy (in terms of representation of the actual physical space). The meshes lack much detail, and there is no way to identify individual objects, like a cabinet or a fire extinguisher. They resemble how it would look if a blanket covered the room, as seen in Figure 6.2a. In addition to not being very detailed, the imported meshes are extremely non-performant with a high amount of excess vertices, and oddly constructed faces that can make it difficult to apply proper textures and lighting information in Unity. Therefore, the point cloud and the converted mesh were used as guides for modeling the turbine in Blender. The converted meshes were imported into Blender to get proportions and placement of objects correctly. Autodesk Recap was used to visualize the point cloud and served as a reference while working, where the point cloud could be navigated and viewed in more detail with proper textures and colors. This process is a meticulous one but results in a model that can include a desired amount of detail and interactable objects, while still being performant enough for VR. It also allows for more flexibility, as the model can be changed and improved on later. The finished model of the tower base can be seen in Figure 6.2b. For each created component of the turbine, this process of gathering necessary data and information, modeling the component, then applying textures and other visual effects had to be done.



(a) Mesh converted from a point cloud, imported in Blender. Depicts the base of the wind turbine tower.

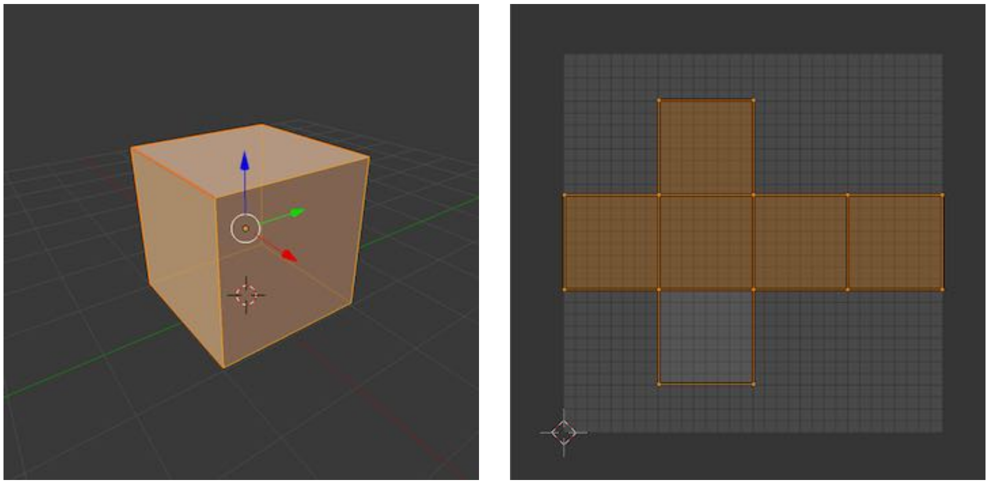


(b) The model created in Blender using the scan mesh as a guide.

Figure 6.2: The tower of the wind turbine, modeled after the 3D scans from TrønderEnergi.

UV-mapping

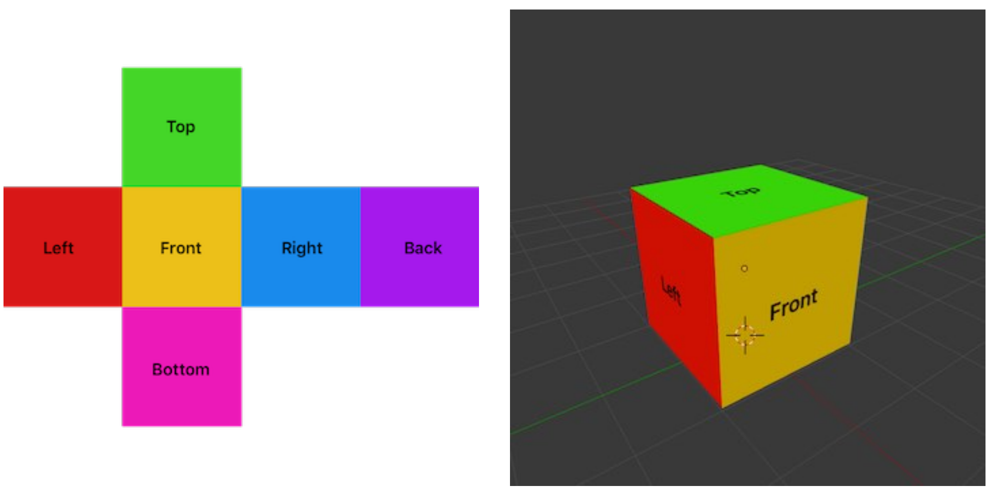
To create a realistic environment, models will need more details than what the geometry and the default gray colors of the modeling software provides. The first step towards providing more realism is to create UV-maps for each model, since a UV-map is used for applying textures, normal maps, height maps, and proper baked lighting. An UV-map is a mapping between the 3D model, and a 2D representation of that model (Blender Foundation 2019b) so that Unity can apply textures and materials at the correct location on the object. In 3D space, the model has X, Y, and Z coordinates, while in a UV-map it only has the U and V coordinates. Each vertex at (U, V) in the UV-map, maps to a vertex at (X, Y, Z) in the 3D space for the object. The process of creating this representation is called *unwrapping*. Blender provides tools for creating UV-maps for objects, including a tool called *Smart UV Project* (Blender Foundation 2019c), which automatically unwraps the object. The automatic generation works well for some use cases, but particular cases need manual editing, or even completely manual unwrapping in order to have a satisfactory result.



(a) A simple 3D cube in Blender.

(b) The corresponding 2D UV-map of the 3D cube.

Figure 6.3: A 3D cube and its corresponding UV-map.



(a) A texture created in Sketch, using the exported UV-map of a cube as a guide

(b) The texture applied to the 3D cube.

Figure 6.4: A texture drawn on the UV-map shown in 6.3, and applied to the cube.

To take a simple example, the process of UV-mapping and texturing will be explained with a 3D cube. In Figure 6.3a, a representation of a 3D cube in Blender is shown. Its corresponding UV-map can be seen in Figure 6.3b. The term *unwrapping* becomes clearer with this example, as the UV-map looks like the cube, just folded out or unwrapped.

The UV-map will then need to be exported as a picture file, then imported into a image editing tool to create the texture itself, using the UV-map as a guide. In Figure 6.4a the texture drawn upon the exported UV-map can be seen. To illustrate the mapping each side of the cube is colored and labeled to identify it. This is a simple example, but illustrates how the workflow of modeling, unwrapping and texturing an object is done. UV-mapping is also essential for providing information to Unity to apply *baked lighting* to static objects, which is explained in the following section.

Baked Lighting

Baked lighting is a technique that provides high-quality lighting for static objects in the scene, with no runtime overhead (Unity 2018a) other than the memory usage of the light textures. Lighting is pre-calculated, and a *lightmap* for the scene is created and used to provide lighting information onto objects. It is important to note that non-static objects, that is, objects that the user can interact with or objects that will move does not affect or get affected by this generated lightmap. It is therefore important to combine baked lighting with real-time lighting (Unity 2018h). As described earlier, UV-mapping is an important element in providing sufficient information about the object to Unity for baked lighting to look as wanted, as the generated lightmap is essentially a texture map that is applied to the objects in the scene. In Figure 6.5, the wind turbine tower can be seen with both textures, baked lighting, and real-time lighting applied.

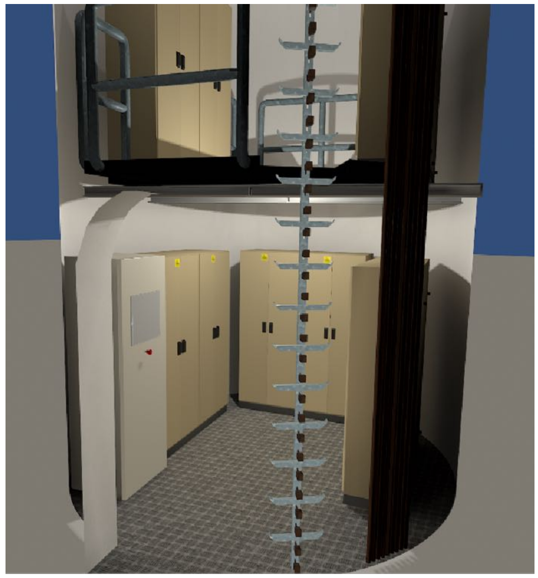


Figure 6.5: The bottom of the turbine tower, with textures and baked lighting in Unity.

Climbing

How is climbing simulated in VR? One possibility is to have a teleport function to appear at the top of the tower instantly. It was experimented with in the early prototyping but quickly deemed as insufficient, as it gave no impression of the actual height of the turbine. The user needs a way to understand that a huge part of the work is climbing around in ladders and understand the importance of security and safety while doing so. Luckily, VRTK includes utility classes for simulating climbing in VR, where the user will be able to grab certain objects, then pull the controller down to drag themselves up. To enable climbing, a *VRTK_PlayerClimb* script needs to be applied to an object in the scene. The script enables moving of the user when the user grabs onto climbable objects (VRTK 2018f). Then the objects that can be held onto by the user needs to be defined, which, as seen in Figure 6.6 is made to be red spheres attached to the ladder steps in the tower. This was done to make them easily identifiable by the user. To mark the spheres as an object that can

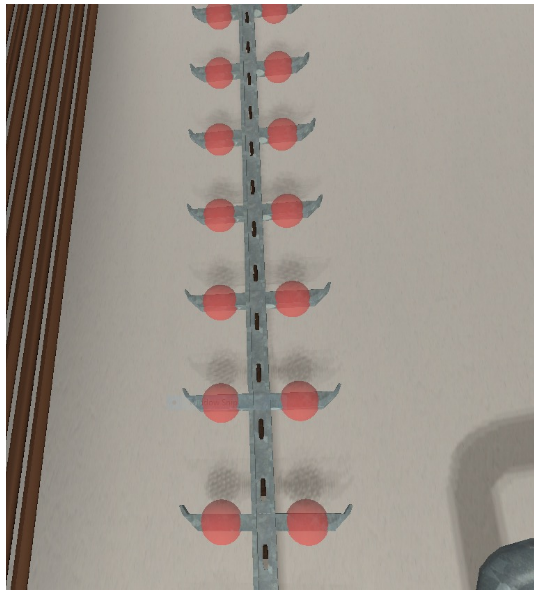


Figure 6.6: The ladder with red spheres that indicate where it can be grabbed.

be used for climbing it needs to be interactable (as described in Section 6.3) and additionally have a *VRTK_ClimbableGrabAttach* attached to it (VRTK 2018b). In this iteration the side grip buttons on the controllers was used for grabbing the climbable spheres on the ladder. No more work is needed to have a functioning climbable ladder! This is one of the cases where VRTK proved extremely useful, and a possibly complex feature was implemented rather fast.

6.5 Creating the landscape

To create a landscape in Unity, specialized *terrain* objects should be used, as these are highly optimized for rendering large landscapes (Unity 2018i). To generate the landscape, data in the form of a heightmap was collected and then converted to a terrain object in Unity. As described in Unity documents, a heightmap is a greyscale image where lighter pixels are higher elevations, and darker are lower (Unity 2018d). Nguyen and Dang (2017) mentions that using heightmaps and converting them to terrain models is a great way to create environments in Unity (Nguyen and Dang 2017). There are two main advantages to this method: There is no need to design and create a believable landscape manually, and the landscape will resemble a real-world location.

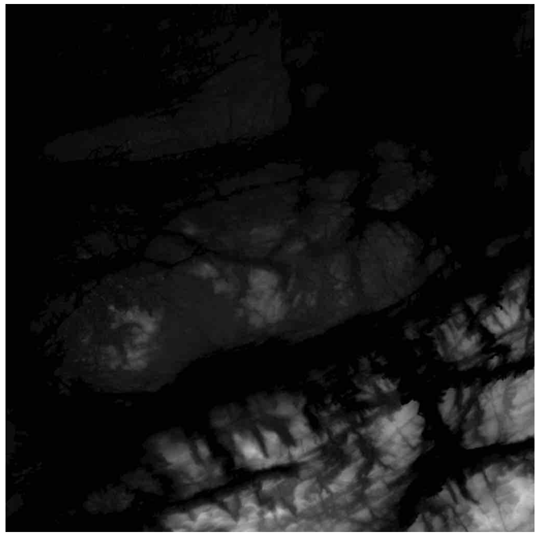


Figure 6.7: Heightmap texture data downloaded from terrain.party, 60km by 60 km. The lighter pixels means more elevated areas. The island in the middle is Hitra, the upper body of land is Frøya, while the lower parts are the coast of Trøndelag's main land.

6.5.1 Collecting and converting height map data

The height map data is collected from `terrain.party`, which lets you choose any place in the world, at a maximum of 60x60 kilometers and download a detailed height map for that particular region. A height map of the island Hitra can be seen in Figure 6.7. This height map was then imported as a normal texture into Unity, and then by using a script by Haines (2012), it was converted to a height map and applied to a terrain mesh (Haines 2012). After generating the terrain, it was resized to 6000 by 6000 units in Unity. As this application is developed as if 1 Unity unit equals 1 meter, which is the recommendation by Unity (Unity 2017), the resulting terrain is a 1/100 of its real-world counterpart as the downloaded height map data represents 60x60 km. The generated terrain mesh can be seen in Figure 6.8.

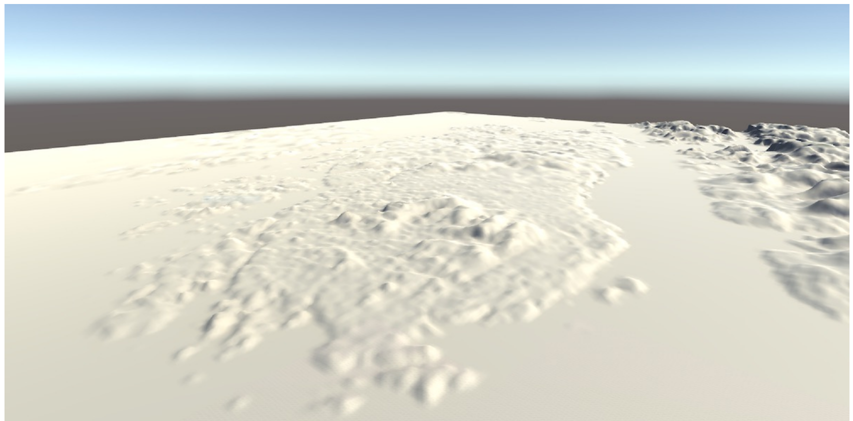


Figure 6.8: The terrain mesh generated from the height map.

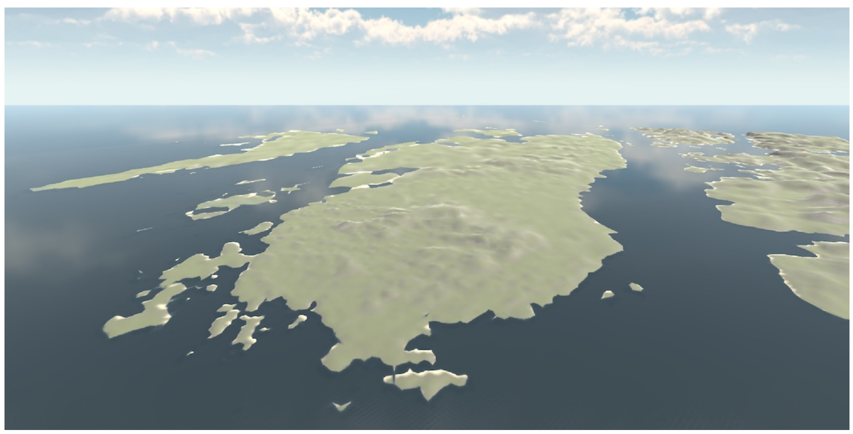


Figure 6.9: The terrain mesh painted with textures, added water and skybox.

6.5.2 Texturing and adding details

A gray terrain is not ideal, nor realistic, thus texturing, adding water and applying a skybox was done. Unity has tools for manually painting terrain textures onto terrain meshes, but by using a tool called Gaia, the process can be automated and therefore made quicker. Gaia can be used to generate terrain meshes, paint them with textures, and add details like water, trees, grass, and rocks (Procedural Worlds n.d.). It uses a set of textures and predefined rules to determine where to paint which texture. Gaia was used to add textures to the terrain and an object representing the water. In addition, a skybox was added for more detail to the world. The finished terrain at this stage can be seen in Figure 6.9.

6.6 Main menu

The main menu consists of a single plane the user stands on, where two large buttons appear in front of the user (see Figure 6.10). Here the user can either choose to initiate the teleport tutorial (Section 6.7), or the application itself. By giving the user a choice, they will not need to complete the tutorial every time they should use the application.

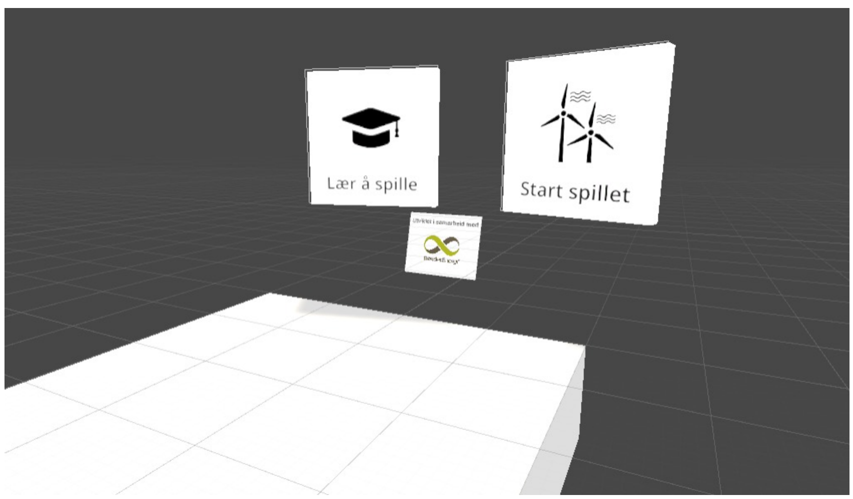


Figure 6.10: The main menu of the application, where the user chooses to either start the application or start the tutorial.

6.7 Teleport tutorial

The goal of the teleport tutorial is to instruct users on how to the teleport functionality works in the application. The user is placed on a tall platform and has to subsequently teleport to a series of platforms to reach the other end of the room. Each platform will be highlighted with a green glow to signalize that the user should initiate a teleport towards this platform. When the user

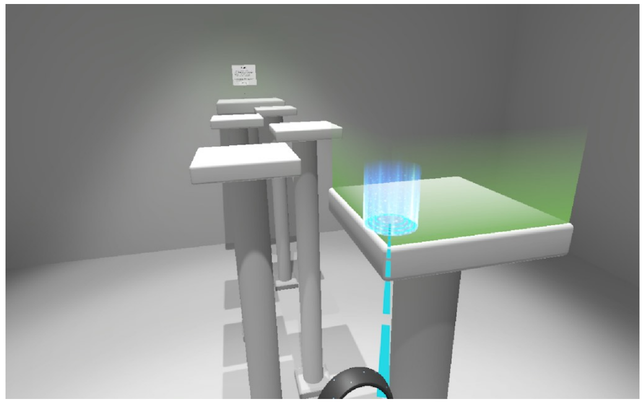


Figure 6.11: The teleport tutorial for the application.

successfully teleports to a platform, the next one will light up. The scene is depicted in Figure 6.11. The tutorial is created as a universal tutorial and should be easily reusable in new virtual internships. The scene is modeled in Blender and directly imported into Unity to apply scripts for virtual reality interaction and teleporting.

6.8 In Game User Interface

The first version contains an in-game menu that lets the player navigate faster in-game. The menu was first designed in Sketch and then necessary assets were exported from Sketch as PNG files and imported into Unity, recreating the menu there with the desired functionality. The menu can be seen in Figure 6.12, with to main options: *restart* or *go to*. *Restart* will reset the game, while *go to*, leads to the navigation menu where the user can navigate to certain areas in the game. Opening the menu is done with the menu button on the hand controllers. Pressing the menu button a second time will close the menu. The menu is spawned at a certain distance from the player in front of the HMD so that no matter what direction the player looks in, it will be visible immediately.

(a) The in game menu.

(b) The navigation menu.

Figure 6.12: The first version of the in game menu.

6.9 Results of testing for this iteration

The first user test on the application was with students at NTNU. The test was conducted at the IMTEL VR Lab at NTNU Dragvoll over two days, 14th and 15th of February. In total, 44 users tested the application. The results presented in this section are both observations done during testing and direct feedback from the testers. The subjects all answered a questionnaire after testing the application. The experience gained from the tests and observations acts as input into the features and fixes for the next iteration, described in Section 7.1.

6.9.1 Observations

During testing, observations were done to understand how users use the application and where improvements are necessary. Specific problems detected during observation are summarized in Table 6.3. The focus of the test was to detect problem areas during the usage of the application, not if users could complete certain tasks. Since no specific tasks related to the job was yet implemented, there was no need to evaluate if these worked. In general, observations show that users struggled with understanding what to do and how to do it. More guides and in-game hints are needed to direct users towards relevant areas and objects.

ID	Observation
O1	Users miss the teleport point in the teleport tutorial to get back to the main menu, resulting in them falling down from the platforms.
O2	Needs more tutorials or in-game guides so users understand what to do, and how to interact.
O3	Climbing is hard for many users. They fall down, and many give up. A security system should be implemented, along with more feedback when climbing.
O4	More robust spawnpoints, especially inside the windmill. Should correct the position of the play area.
O5	The pointer used for selecting teleport location is a bit too arced. Most user teleport too far. Can make it straighter.
O6	Users struggle with finding and understanding the grip buttons on the controllers.

Table 6.3: Observations made during user testing 14th and 15th of February.

6.9.2 Questionnaire

The questionnaire in this test did not focus on tasks for the user to do, just the experience of using the application. The questions used the Likert agreement and rating scale. The results from the questions can be seen in Table 6.4.

The survey did also include three long answer questions to get more details on how the user experienced the application and their thoughts about it. The three long answer questions were: *Q5: Suggestions for improvement of the application*, *Q6: Problems encountered during usage of the application*, and *Q7: Other comments*. The following sections will go through answers received on

Question ID	Question	Average Score	Median Score
Q1	How will you evaluate the visual quality of the application?	3.93	4
Q2	How will you evaluate the quality of the interactions in the application?	4.07	4
Q3	I felt motion sick	1.93	2
Q4	I enjoyed using the application	4.5	5

Table 6.4: The results of the questions handed to participants after trying the application. Tests conducted 14th and 15th of February. Q1, Q2 and Q4 are rated by the Likert Rating scale, and Q3 is reated by the Likert Agreement scale.

each of these statements. The parts of feedback that is inside square brackets are added by the author to interpret the feedback when unclear.

Q5: Suggestions for improvement

The feedback for improvements focused on graphical improvements, more nature, more things to look at, and more things to do. Additionally, one user commented that the teleport function was to *instant*, and it would be more realistic to see the whole travel of the teleport action. Simulating the whole travel could help users understand better what is happening. Observations show that many users will become disorientated when they one moment are standing somewhere, and suddenly, the next moment finds themselves standing at a different location. In tight spaces, like inside a wind turbine, this effect seems to be even more detrimental to the experience. It was also mentioned that the grip buttons are too hard to use. This difficulty was also observed, and a possible solution will be not to use them at all but instead use the triggers for all the interactions. Following is the list of feedback received for Q5 in no particular order.

- It would be fun if, when teleporting, it looks more like Mario where you join in and see the whole teleportation. I felt it snapped from one picture to another instantly, and I think the experience will be more realistic if the user gets to see the "fall".
- Make alternatives more clear, in terms of how you do things. [The user means that it can be hard to know what to do next]
- If people feel motion sick I think you can include a nose to reduce this effect.
- If you could use the legs when climbing it would have made it better, but that is probably hard. You can have more feedback when falling, so it is experienced more.
- More details at ground level, so the difference between the top and bottom of the ground is more distinct.
- More interactions in the wind turbine.
- Make the "grip" button more clearly marked.
- Better graphics at the top, more to look at?
- It is in the beginning, but generally focus on graphics and user-friendliness and tutorials.
- More things to do?
- Tasks on the way as you climb, more challenges.
- Graphical improvements
- Some improvements in the button layout of the controllers. I felt it was hard to use the side buttons [Grip buttons]

- Haptic feedback when the ladder is gripped, optionally sound.
- Names on places seen in the environment. Information about the surroundings.
- Trees outside and more nature.
- See the engine inside the wind turbine. [User here means the generator and machinery]
- Would be cool to have a machine room and more technical parts in the turbine.
- More details in the landscape.
- Real scenery photographs at the top.
- Possibility to change the wingspeed on the turbine [Would simulate a change in the wind speed]. Something to set things into perspective [Nature, trees, and other objects to see the actual size of things.]. A little unusual to use the side grip buttons but made sense after some time.

Q6: Problems encountered during usage

The general feedback on problems encountered is that teleporting and climbing the ladder can be finicky. Especially climbing was observed as a problem during the tests.

- Took some time to understand the controllers, but after that it was fun.
- Sensitive teleporting and the instructions was very close.
- Could not teleport to the top of the turbine even though i climbed all the way to the top.
- Hard to get used to the game, but that is more about the gamer than the game.

- Easy to fly up the ladder
- Problems when climbing the ladder, hard to grip the steps.
- Easy to fall down when climbing inside the turbine.
- Hard to climb up the ladder, but could be my skills.
- I was inside walls sometimes.

Q7: Other comments

- Fun concept and the height felt real. Fun to try and good instructor.
- Should have a problem or typical task to complete on a turbine and get instruction on how to do it.
- Useful with the blue grid showing the real-life walls.
- Hard to remember all the buttons on the controller.

Chapter 7

Development - Second Iteration

This chapter will describe features implemented and problems that occurred in the second iteration of development. This iteration lasted from 19th of February to 15th of March.

7.1 Features and Fixes

Implemented features and fixes are represented in Table 7.1, with an ID and description. The following sections in this chapter will describe implementation details for some of the features. This second iteration included both fixes and features based on results of the user testing after the first iteration (6.9) and necessary and planned features for the application. Not all fixes and features require a separate section to explain how and why, therefore, not all entries will lead to a section in the table. In addition to the features and fixes described in the table below, Section 7.7 describes a recurring problem with committing changes to Git properly and how it was solved.

Table 7.1: The features and fixes for the second iteration of development.

ID	**Description**	Relevant Sections
F2.1	Move the platform in the teleport tutorial backwards to make it harder to miss it when trying to end the tutorial	None
F2.2	Fix the spawn point inside the tower so that the user spawns by the door, no matter their position in the play area.	None
F2.3	Use TextMesh Pro for UI text to make it more readable.	None
F2.4	Spawn the menu closer to the user if it collides with objects.	7.6
F2.5	Adjust the size of the in-game menu depending on the distance it spawns from the user.	7.6
F2.6	Change the teleport technique so that it is visible that the user dashes to the selected location.	7.2.4
F2.7	Adjust the teleport arc to be less arced, to make it easier for the user to coordinate the controller movement with the actual point they are pointing at.	None
F2.8	Change to use to trigger button instead of the grip button for climbing.	7.2.2
F2.9	Implement security line on the ladder so users can secure themselves and hinder them from falling when climbing.	7.2.2
F2.10	Make additional feedback and guidance on the ladder. More colors and haptic feedback to indicate states when climbing.	7.2.2

Continuation of Table 7.1		
ID	**Description**	Relevant Sections
F2.11	Textual and visual guides towards objects of interest.	7.2.1, 7.2.3
F2.12	Employ Level of Detail for performance-intensive objects.	7.3
F2.13	Create an exterior model of the wind turbine that corresponds to the turbine that TrønderEnergi has supplied scans for.	7.4
F2.14	Create a task turning off the rotor in the turbine	7.5
F2.15	Create a task to climb the ladder	7.5
F2.16	Create a task to change the signal lamp light bulb at the very top of the wind turbine.	7.4, 7.5
F2.17	Rework the in-game menu to better support an overview of tasks and completion of these.	7.6
F2.18	Add more details to the terrain: trees, rocks, a car, and a small town to view from the top of the turbine.	None

7.2 Improving guiding, user interactions and feedback

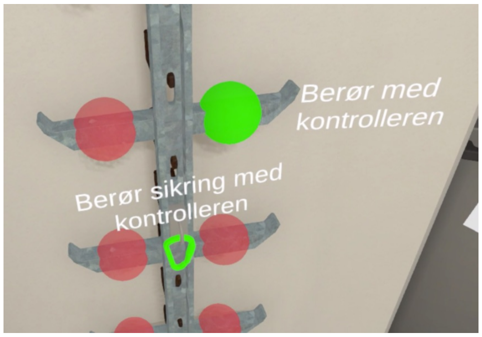
The first iteration of the development did not implement any features that focus on guiding and helping the user to understand how to do things. The second iteration adds more features that will aid the user to understand what to do and how to do it.

7.2.1 Interactable guides

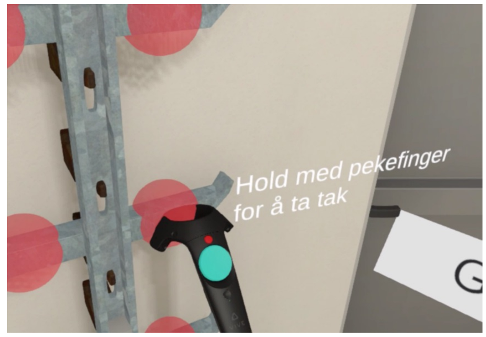
A generic and reusable object acting as a guide to an interactable object was created. This guide, consisting of a script and a text label, will be attached to an interactable object or linked by referencing the interactable object. As the user interacts with the object, the guide object will receive events and change the text accordingly. For example, a guide for the rotor-lock button will initially say: “Touch the button with the controller”, then when the button is touched, the text can change to “Pull the trigger”. This way, the user does not have to read a wall of text when interacting with an object, and the necessary information is available when needed.

7.2.2 Ladder guide and usability

The ladder previously included no guiding, or affordance of what to do when standing in front of it. Several means were used in order to make it more obvious to the user how they should climb the ladder. The first step was to abandon the grip button in favor of the trigger button. By removing one button, the number of buttons that the user will need to remember is reduced to three: trigger, touchpad, and menu button. Second, the spheres, indicating where to grab the ladder steps were changed so that when the controller touches a sphere, it changes the color to blue to indicate that the controller is near enough to grab the step. When the step is grabbed, the color will change to green, to indicate that the user has a hold of that step. Additionally, haptic feedback was added, both on touch and use of the sphere so the users can *feel* that they are grabbing it. Third, an interactable guide (described in Section 7.2.1) was added to the first few steps that tell the users to touch the sphere, grab it by pulling the trigger, then drag themselves up. Fourth, in combination with the interactable guide, an arrow object was spawned by the users' controller to show that they must pull the controller down, in order to drag themselves upward in the game.



(a) The object hints hinting that the spheres are grabbable.



(b) The label changes text to tell the user to hold with the trigger to grab the step.

(c) When grabbing, the sphere turns green and an animated green arrow will hint the user to pull down.

Figure 7.1: Implemented features that help the user understand how climbing is performed.

An additional feature that is both realistic and useful in the game is the security system for the ladder. The user can fasten themselves with a carabiner hook to the ladder. By doing this, they secure themselves in such a way that they will not fall if they let go of the ladder. From there they can continue their climb upwards to the top. Security on the ladder is both realistic and practical, as many users experienced falling before reaching the top of the turbine and therefore giving up and resorting to just using the navigation menu to get to the top.

7.2.3 Object pointers

A green, animated, bumping arrow was added to make it clear where things are. It animates slowly up and down to gain the users' attention, and has a huge text label on top describing what is located at that particular location. It was used to mark the location of the entrance of the correct wind turbine where the user is supposed to conduct maintenance.

7.2.4 Dash teleporting

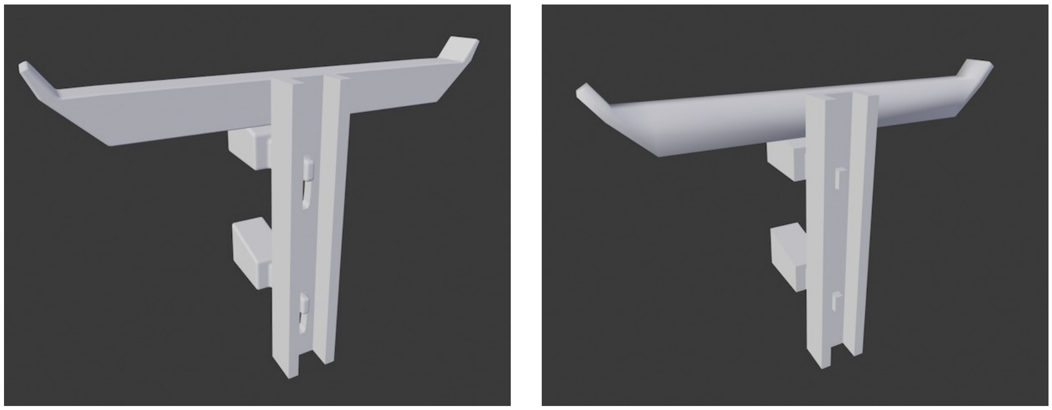
To give the user more feedback when teleporting, and to make them less disorientated when doing it, the standard *flash* teleporting was exchanged with *dash* teleporting. Instead of the user suddenly appearing at a new location in the game when the teleport action is initiated, they will dash, so that the application will visualize the travel by quickly moving the user along a straight line towards the target destination. This was done by exchanging the current locomotion script described in Section 6.3 to *VRTK_DashTeleporter* which updates the user's position with lerp (linear interpolation) to the destination position (VRTK 2018c). It is important that the travel is fast enough, above some threshold, so that the user does not experience motion sickness when teleporting. Possibly, in the future, this could be implemented as a preference that the user can change themselves, if they prefer the flash or dash teleporting.

7.3 Level of Detail

Level of detail (LOD) is a technique for optimizing rendering of detailed objects at different viewing distances, by reducing the level of detail of an object as the viewing distance increases, and vice versa (Unity 2018e). Maintaining a comfortable, stable framerate is crucial for the experience in VR. In this section, it will be demonstrated how a technique using LOD for objects can improve the performance of the scene. In this case, the interior of the wind turbine, specifically the ladder, will be used as an example.

The original ladder consisted of fairly high-resolution elements stacked upon each other to form the complete ladder. In Figure 7.2a, the original ladder element can be seen. That element consists of 1,444 vertices, 2820 edges, and 1380 faces. These are not extreme numbers, but this single element will be stacked upon each other many times, and the machine will have to render many instances of the model. In Figure 7.2b a lower poly version of the same ladder element can be seen. It has sharper edges and less detail but still represents the ladder in an acceptable manner. This version consists of just 76 vertices, 136 edges and 68 faces! That is much less detail for the machine to render with each pass. Instead of using this low detailed version in place of the detailed model to increase the performance, LODs should be used to keep both the performance and visual quality at acceptable levels. Since the user climbs the ladder, they will have it close to their eyes, and the boxier low detail model would not induce much realism. LODs take advantage of the fact that the user does not need to see the maximum detail of every object at all times. As objects are further away, the detail level of the geometry can be reduced without noticeable visual impact. The high-quality model is used on the ladder elements that are closer to the user, and lower detail further away. As the user climbs up the ladder, the elements that come close to the user will have their detail level increased.

Dynamic LODs can be achieved in Unity with the use of LOD groups. The two objects will be exported from Blender, into Unity, then defined as different LODs for the same object. The high-quality version will be automatically swapped out with a lower quality one, if the user is further away. The definition of a model's LOD levels can be done automatically by Unity if the models are named in this scheme: *ModelName_LOD0, ModelName_LOD1, ..., ModelName_LODn* and so on up until the desired amount of LOD levels *n*. Additionally, if the **.blend** file is imported into Unity directly, all these versions of the object can be contained in the same file, named with the scheme



(a) The orginal ladder element.

(b) The ladder element with a reduced polycount.

Figure 7.2: Demonstrating LOD technique with the ladder elements of the turbine tower. In Unity the different levels of detail can be transitioned between depending on the distance between the user and the object.

described above, and Unity will automatically apply a LOD Level Group component to the model object, enabling LOD levels for that object. The lowest LOD level is the *culling* level, which is the distance when the object will not be visible at all. The culling level can be useful in large scenes with many details and objects. A typical use case for culling is, for example, to cull grass in larger landscapes, as displaying grass over large areas will be expensive.

7.3.1 Performance Impact

There is quite some extra work needed to create several versions of an object, therefore the impact should be noticeable in order for it to be worthwhile. In Table 7.2 a comparing of rendering data as the user stands at the bottom of the ladder and looks upwards, with and without the use of LODs. It has a massive impact on the total number of triangles and vertices in the scene. The FPS measurements should not be regarded as the actual performance of the application, as they are measured in the Unity editor and not in a complete build of the application and therefore includes some overhead. The main point

LOD	FPS	Triangles	Verts
No	50-90	9.6 M	22.7 M
Yes	85-90	2.2 M	1.9 M

Table 7.2: The impact on FPS and rendered triangles and vertices with and without using LODs.

is that it can be a valid strategy to introduce LODs for objects to improve performance and reduced the number of triangles rendered vastly with next to no visual impact. This example was just one object, and it improved the scene noticeably, imagine having hundreds of detailed objects and the amount of performance that could be gained with this technique! Therefore, in the further development of this application, the LOD technique is used to keep the performance at a satisfactory level for VR. LOD is something that should be stressed for all future virtual internship applications as bad performance can be hugely detrimental to the experience in VR.

7.4 Wind turbine exterior model

In the previous development iteration, a 3D model fetched from the internet was used as the exterior of the wind turbine. This model was not consistent with the interior created from the 3D scans. The scans from TrønderEnergi were from an Enercon wind turbine, so the exterior of this turbine was recreated in Blender in the second iteration. The 3D scan point cloud, along with pictures of the turbine, guided the design. Additionally, in order to get the height and proportions correct, the complete mesh object converted from the 3D scans were imported into Blender and used as a guide. See Figure 7.3 for the model of the turbine in Blender. Figure 7.4 depicts the exterior model of the turbine added to the Unity scene with the terrain object. Along with creating the new model, more turbines were added to the scene since usually, there is several turbines clustered together in what is termed as a wind park.

Figure 7.3: The model of the Enercon wind turbine created in Blender. This file is directly imported into Unity.

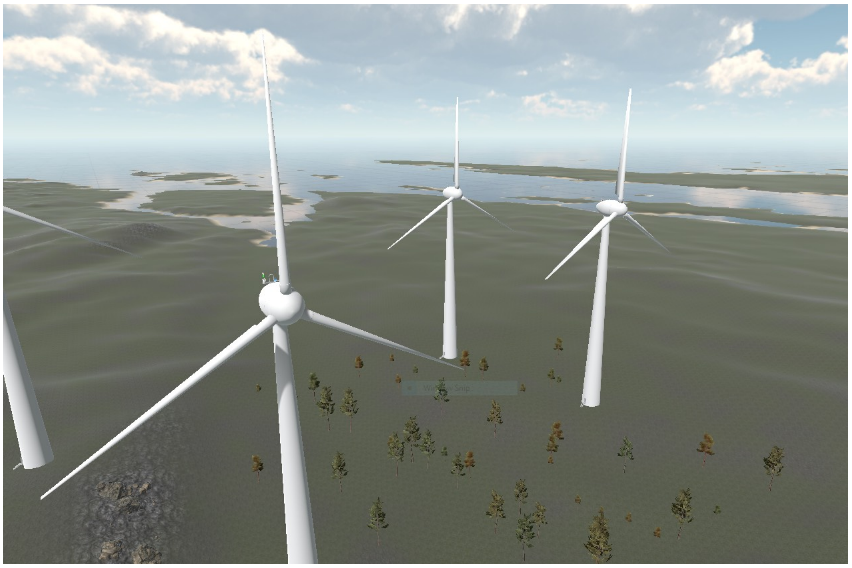


Figure 7.4: The turbine models added to Unity.

7.5 Tasks

The second iteration focused on adding tasks to complete in the application. Three main tasks was added: *turn off the rotor*, *climb the ladder* and *change the signal lamp light bulb*. As the climbable ladder was already implemented, a trigger at the top was added to detect if the user climbed the ladder. Turning off the rotor is implemented at the bottom of the tower on the control cabinet. By pushing a red button on the cabinet, the user completes the task, stopping electrical production and activates the rotor lock system in the turbine. On a screen, at the cabinet, the user can see the current status, output, and RPM of the turbine. These values change as the button is engaged. With the new exterior model described in Section 7.4 a signal lamp that sits on top of the turbine to warn air traffic about the turbine's position, was added to the model. A task to change the light bulb in this lamp was then added, where the user has to open the lamp, take out the old bulb and retrieve a new bulb from their inventory (see Section 7.6 for a description of this menu.), and place it in the lamp.

7.6 Improving the In-game menu

The second iteration added more details to the application, and as a side effect, the in-game menu needed an upgrade and more detailed pages. The increased need for details resulted in a redesign of the menu to make it easier for users to understand and navigate the menu. The redesign is based around tabs, like in a web browser, where each tab represents a part of the menu: *tasks*, *navigation*, and *gear* as seen in Figure 7.5. The menu system is fairly configurable and would be easily reusable in another application. Each *tab* in the menu is essentially a prefab, with a title and a tab content view. A prefab in Unity is a pre-made object, complete with model, components, property values, and children (Unity 2018g). When the user activates a tab, it is asked to render its content and will gain control of the content view of the menu. Adding a new tab is done by creating a new prefab, with a title and a content view that adhere to these requirements, then register the new tab in the game menu controller object. This way, the component should be easily reusable and extendable for other virtual internships. The currently active tab in the menu is colored yellow, and buttons in the menu will become highlighted with a bright blue color to indicate that the user can select a button or element by pressing the trigger button on the controller.

To give the user an overview of all the tasks, and what tasks require completion, the menu contains a page with a list of all available tasks. The list of tasks can be seen in Figure 7.5a, where each element in the list is a task. Each task has a title, a status (completed/not completed) and a button that takes the user to a new page with more information about the task (see Figure 7.5b). The page with more information describes in more detail what the user is supposed to do, useful tips and the location of the tasks.

(a) The redesigned in-game menu, with tabs at the top for each page. This shows the list of tasks the user should complete.

(b) When the user select ”Informasjon” on a task they are taken to the detail page that explains more about the task.

Figure 7.5: The task tab of the redesigned in-game menu system.

The gear tab is where the user can find needed objects to complete tasks. In the current state of the application, only one piece of gear is needed (a lightbulb), but in the future, it can be imagined that more items are necessary. The inventory is a simple list of items the user has, and by pressing the *use* button on an item, it is spawned in the user's hand (controller).

The navigation page stayed mostly the same as for the previous iteration, with four plain buttons where the user can choose where to go in the application, as an alternative to teleporting and climbing.

7.6.1 Spawning the menu

The previous iteration of the menu spawned at a predetermined distance with a specific scale in the direction the user was looking. The fixed distance resulted in that the menu would often open inside cabinets or behind walls if the user were looking at objects while opening the menu. Along with the redesign, the spawning system was improved to spawn at a dynamic distance from the user, depending on the number of objects in the user's field of view. If the user is standing in front of a wall, a raycast test is performed to detect the wall; then the menu is spawned closer to the user in order not to be obscured by the wall.

Additionally, the scale of the menu is decreased or increased depending on the distance it is spawned from the user. This way, the eligibility is not impacted when spawning at varying distances. A minimum and maximum spawn distance is, however, necessary as spawning the menu to close to the HMD position resulted in unintentional artifacts, like two menus at the same time, and an uncomfortable experience. Spawning the menu at a too large distance was just deemed unnecessary as it would eventually become so large that it would clip into objects. The spawning system is decoupled from the menu system itself so that it is an easily reusable component that can be used for any menu in this application or other applications.

7.7 Problems with commiting changes to the terrain

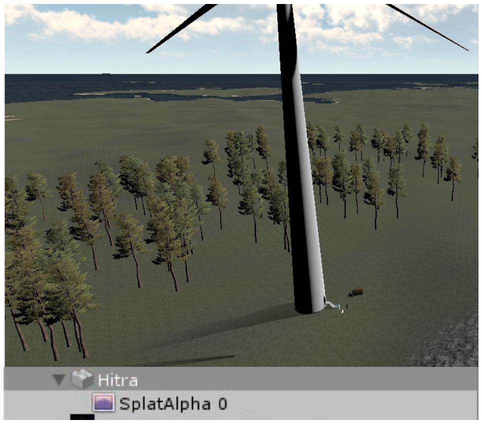
A notable problem with using Git and Unity is that committing and merging files is not always behaving as expected. In this section, a problem encountered when committing changes to the terrain file in Unity is presented, along with a solution.

7.7.1 Demonstrating

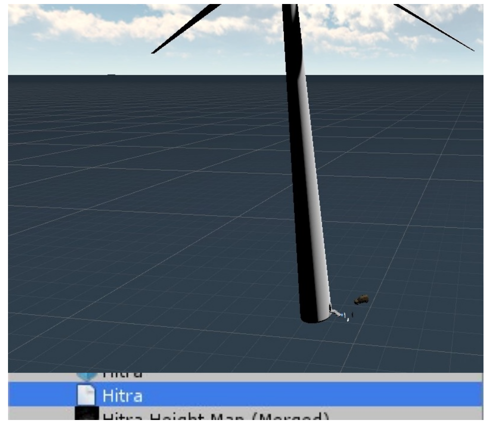
This part will demonstrate the problem briefly. If there are two branches, *t1* and *t2* that is on the same head commit, and the currently active branch is t1, then each branch contains the same terrain data file as a source for their terrain mesh object. In Figure 7.6a the terrain file is in working order as shown in the Unity editor. Say a developer then applies some changes to this terrain, for example, add some trees, paint it with different textures or adjust the elevation of areas in the terrain, then commits these changes, which in this case would be visible for Git as a modification to *Hitra.asset*. A normal workflow in Git is to merge work into existing branches, merging t1 into t2 should be no problem as t1 only contains one new commit (with the terrain change), compared to t2. After merging, if loading the project in the Unity editor, the terrain will disappear, and the terrain data file will be broken (see Figure 7.6b)

7.7.2 What created the problem and how to solve it

In Unity a terrain consists of two files: the *data* file and the *terrain* object file. When editing the terrain in the Unity editor, this will edit the underlying data file. The problems stems from the configuration of Git with Git Large File Storage (LFS). The configuration is done through *gitattributes* that tells Git how to handle large files and how to serialize and merge them. Since Unity, in general, can force storing Asset files as text files, it is normal to have a rule that uses *merge=unityyamlmerge* in the attributes for these files. Now



(a) How a working terrain data file looks in the Unity Editor file inspector and scene.



(b) How the terrain data file looks like when it is corrupted on commit in Git.

Figure 7.6: Demonstration of a broken terrain data file. The terrain will disappear in the scene when the data file is corrupted.

there are some exceptions to this, where Unity stores asset files as binaries even though force text serialization is on, and Git will wrongly interpret this as text files inserting needed line ending changes, and breaking the file. Therefore, an exception is needed for files that use the *.asset* extension, but stay as binary files even though forced text serialization is activated. By applying exceptions to the terrain files, this problem will no longer happen.

7.8 Results of testing for this iteration

The first test of the second version was conducted at the VR Lab at Dragvol 11th of March with 5 NAV employees from Moss testing the application. The second test was done at the VR Lab at Dragvoll at the 15th of March with eight users; five of them where job seekers in the NAV system and where 3 NAV employees. The last test conducted on this version was done 20th of March with six students from upper secondary school attending an event on NTNU. The results are separated as the test groups were different, and so were the questionnaires handed to the distinct groups.

7.8.1 Observations

This section will briefly explain observations made in all of the tests. The increased focus on guiding within the application seems to help the users quite a bit. Especially climbing is learned quicker than in the previous version, where much explanation was needed. Now the application takes the user through the process interactively.

ID	Observation
O1	Some users struggle to hit buttons in the menu, could be that the pointer is a little large compared to the buttons.
O2	Signal lamp is too hard to open
O3	People teleport easily into cabinets at the bottom of the tower.
O4	People frequently teleport down to the ground when standing on the top of the turbine without meaning to.

Table 7.3: Observations made at the tests in the second iteration of the development.

7.8.2 Test with NAV employees 11th of March

The first round of users for this iteration tested the application at the VR Lab at Dragvoll. The users were all NAV employees from Moss. After testing the application, the users were handed a questionnaire. As these users had a different stake in the application development, the questions differ slightly from the questionnaire used in the user testing conducted in the first iteration. In total, five users answered the questionnaire, with a median age of 49, and three women and two men. The results from the questionnaires given after these tests can be seen in Table 7.4.

Table 7.4: Results from the questionnaire given to NAV employees that tested the application at the 11th of March at the Imtel VR Lab Dragvoll.

ID	**Question**	**Average Score**	**Median Score**
Q1	How will you evaluate the visual quality of the application?	4.2	4
Q2	How will you evaluate the quality of the interactions in the application?	4.2	4
Q3	I felt motion sick	2.2	2
Q4	I enjoyed using the application	4.8	5
Q5	The tasks I were given felt realistic	4.2	4
Q6	After completion I had a better understanding of the work	4.2	4
Q7	I want to use such application in my work with unemployed	4	4
Q8	Such applications can introduce user for workplaces and typical tasks at the workplace	4.8	5
Q9	Such applications can increase a user's motivation towards applying for a job	4.8	5
Q10	Such applications should be a part of NAV's service to unemployed	4.6	5
Q11	Such applications should be a part of career guidance at schools	4.8	5
Q12	I think I would need help from a technical skilled person to use these applications as part of NAV's service to users	3.4	4

Continuation of Table 7.4			
ID	**Question**	**Average Score**	**Median Score**
Q13	It would be easy to integrate such applications as part of NAV's service to users at my workplace	3.4	4

Q14: Suggestions for improvement

- Sound effects, wind, machines to make it more realistic
- More tasks to complete

7.8.3 Test with NAV users and employees 15th of March

At the 15th of March, another test was held at the IMTEL VR Lab at Dragvoll, with a total of eight users, three NAV employees, and 5 NAV users. The results of the questionnaire with questions related to the application can be seen in Table 7.5.

Table 7.5: Results from the questionnaire given to NAV employees and users that tested the application at the 15th of March at the Imtel VR Lab Dragvoll.

ID	**Question**	**Average Score**	**Median Score**
Q1	How will you evaluate the visual quality of the application?	4	4
Q2	How will you evaluate the quality of the interactions in the application?	3.75	4
Q3	I enjoyed using the application	4.13	4
Q4	The tasks I were given was simple to complete	4.25	4
Q5	There was a lot that seemed illogical and unpredictable	1.86	2

Continuation of Table 7.5			
ID	**Question**	**Average Score**	**Median Score**
Q6	I felt motion sick	2	2
Q7	The tasks I were given seemed realistic	3.5	4
Q8	I was in need of instruction when using the application	3.25	4
Q9	After completion I had a better understanding of what the job is like	3.25	3.5
Q10	Time limits on tasks would have made the application more interesting	3	3
Q11	A possibility for cooperation would have made the application more interesting	3.88	4

Q12: Suggestions for improvements

- Sounds
- More tasks, maybe a video displaying some of the work

Q13: Other comments

- Possiblity for worse weather
- Tools and gear attached to trousers etc.
- Actually felt scared by the height. The application seems realistic, but maybe not a realistic experience in regards to the physical strain in the work?
- I want to know more about what kind of knowledge is needed in the line of work.

After the test session an informal discussion with the users was held, following are some relevant main points that showed up:

- Using these applications is better than reading about the workplace.
- The application had concrete, clear tasks.
- Felt the height.
- To few tasks.
- Movies in combination with tasks are good (like in FiskeVR).
- Maybe important to see the bad sides of a workplace. Bad weather and other conditions.

7.8.4 Test with pupils 20th of March

At the 20th of March, a test with pupils was held, where people still in school aged 17 to 18 (median age 18) attended. In total, there were six users with three women and three men. Table 7.6 shows the results from the questionnaires of the test.

Table 7.6: Results from the questionnaire given to pupils that tested the application at the 20th of March at the Imtel VR Lab Dragvoll.

ID	Question	Average Score	Median Score
Q1	How will you evaluate the visual quality of the application?	4.34	4
Q2	How will you evaluate the quality of the interactions in the application?	4.34	4.5
Q3	I felt motion sick	2.5	2.5
Q4	I enjoyed using the application	4.83	5
Q5	I needed a lot of instruction when using the application	3.17	3

Continuation of Table 7.6			
ID	**Question**	**Average Score**	**Median Score**
Q6	There was a lot that seemed illogical and unpredictable	2.17	2
Q7	The tasks I were given seemed realistic	4	4
Q9	After completion I had a better understanding of what the job is like	3.83	4.5

Chapter 8

Development - Third Iteration

This chapter will describe features implemented and problems that occurred in the third and last iteration of development. This iteration lasted from 18th of March to 26th of April.

8.1 Features and Fixes

The specific features, or implemented fixes, is represented in Table 7.1, with an ID and description. The following sections in this chapter will describe implementation details of the features implemented in this iteration. This third iteration will include both fixes and features based on results of the user testing in the first iteration (Section 6.9) and the second iteration (Section 7.8) and necessary and planned features for the application. The reason the results from the first iteration is mentioned here is that experience and results from the tests there is still relevant in this iteration. Not all fixes and features require a separate section to explain how and why, therefore, not all features will lead to a section in the table.

Table 8.1: Features and fixes implemented in the third iteration of the development.

ID	Description	Relevant Sections
F3.1	Add a scene to display the internals of the nacelle in the turbine, which is an essential part of the workplace	8.2
F3.2	Add a task to inspect and fix electrical cables	8.3.1
F3.3	Add a task to fill oil in the nacelle	8.3.2
F3.4	Add a task to turn on service mode in the wind turbine	8.3.3
F3.5	Move rotor lock task to the nacelle	8.3.3
F3.6	Add more details to the environment and nature	8.4.1
F3.7	Add post processing effects to improve graphical quality	8.4.2
F3.8	Add a system to maintain a score for the user	8.5
F3.9	Add 360-degree video from the top of a real wind turbine	8.6
F3.10	Add 360-degree video where a wind turbine technician explains the control cabinet in the nacelle	8.6
F3.11	Add 360-degree video where a wind turbine technician explains the transformation cabinets in the bottom of the turbine.	8.6
F3.12	A new navigation menu tab that includes a drawing of the wind turbine	None

Continuation of Table 8.1		
ID	**Description**	**Relevant Sections**
F3.13	Make the menu pointer smaller, making it easier to point at buttons.	None
F3.14	Deactivate teleporting from the top of the turbine down to the terrain object.	None
F3.15	Make the pointer for the menu smaller.	None
F3.16	Change the way the signal lamp is opened, from grabbing and dragging to a simple touch and press with the trigger.	None
F3.17	Add extra colliders on cabinets in the tower to block teleportation into them.	None
F3.18	Utilize illegal teleport zones, so users do not teleport to undesired locations.	None

8.2 Nacelle

The nacelle of the wind turbine is the last missing main piece of the construction of the turbine. The nacelle sits at the top, connected to the tower and houses the generator, gearbox, control systems and motors to turn the nacelle and rotor towards the wind direction. The rotor of the turbine connects to the nacelle. It is a vital part of the turbine and important to include in the application. The meshes resulting from the 3D scan of the turbine is used as a guide to model in Blender (see Figure 8.1), along with viewing the scan in Autodesk Recap just as for the internals and exterior of the tower. The point cloud data visualized in Recap can be seen in Figure 8.2 alongside the finished 3D model with textures and lighting in Unity. The nacelle includes three tasks: turn on the rotor lock, fill oil, and inspect cooling fans in the control cabinet. More detailed descriptions of these can be found in Section

8.3.

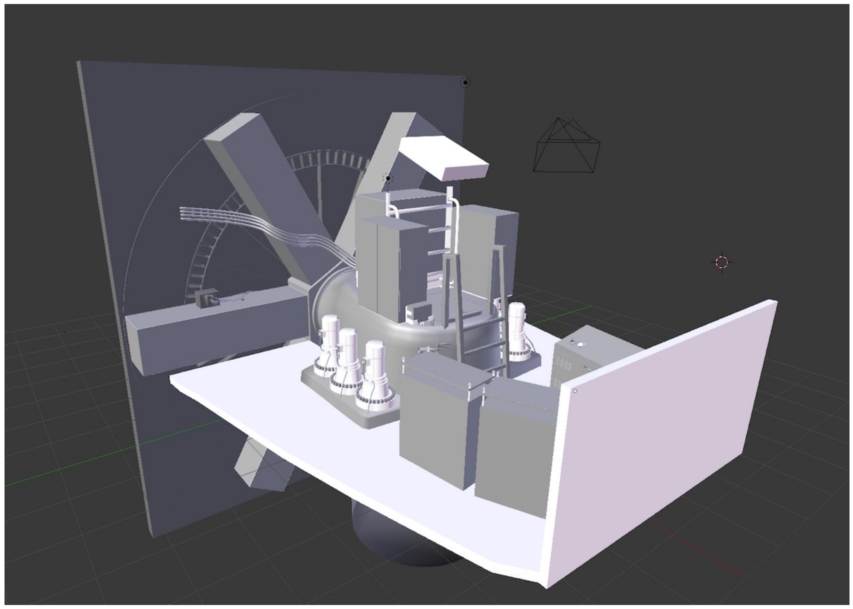
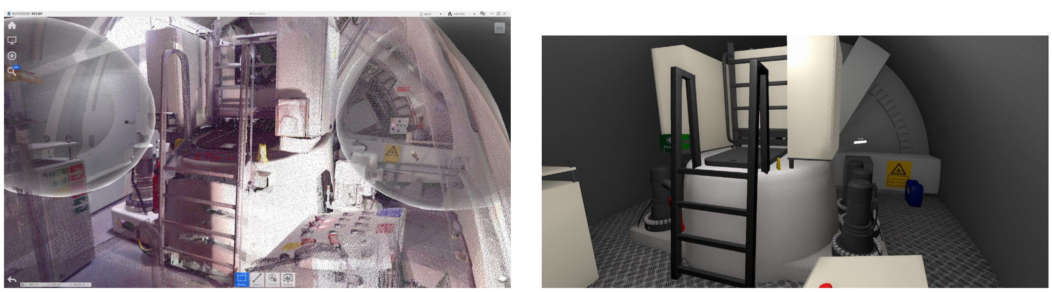


Figure 8.1: The nacelle scene modeled in Blender. The blender file is imported directly into Unity.



(a) The point cloud of the nacelle visualized in Autodesk Recap.

(b) The created model imported into Unity with textures and lighting.

Figure 8.2: The point cloud and the created model of the nacelle.

8.3 Additional Tasks

More tasks were added in this iteration, to give a complete impression of the work, as this is a regular request feature from users.

8.3.1 Electrical cables

In a wind turbine, there are a lot of electrical components, including a generator, converters, sensors, and control systems (Windpower Engineering 2011). A task including some inspection of electrical cables was therefore deemed appropriate. The task is for the user to inspect all the electrical cables in the cabinets at the bottom of the wind turbine tower and if they find any damaged cables, they will need to replace them.

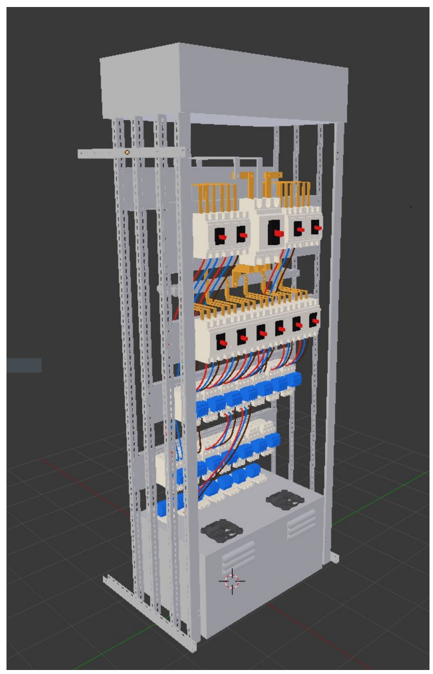


Figure 8.3: The internal components of a cabinet, modeled in Blender. It is based on a model by Ngu Nguyen from Grab CAD: `https://grabcad.com/library/electrical-cabinet-8`.

First, the internal components of the cabinets were modeled in Blender. This was done by first finding a CAD model on the internet of internal components in an electrical cabinet. Further, the model was modified to fit the use case, and the cabinets in the wind turbine. The finished model can be seen in Figure 8.3. Broken cables are identified by creating two versions: one whole and one broken. The broken versions had some of the insulation teared up. In Unity, either the broken or whole version was activated depending on whether the cable should be broken in that cabinet or not. When testing out in VR it was quickly realized that the cables were too hard to identify, so some measure to make the user identify the broken cables must be applied. To do this, realism had to be traded for usability; a flashing color was added along with sparkles flying out of the cable. A trade-off in realism was deemed necessary in order not to frustrate the user by making it almost impossible to identify the broken cables. To fix the cables, the user is required to touch it and press the trigger button on the controller, and the broken cable is switched out with a working version.

8.3.2 Fill oil

The mechanical systems in the turbine use grease or oil to stay lubricated. In order to simulate the process of filling oil, a simplification was done: an oil reserve tank was added to the nacelle scene, along with an oil can that the user can pick up and pour into the reserve (see Figure 8.4). Above the oil tank, an oil meter can be seen so that the user can monitor the level and know when it is full. The oil can is modeled in Blender, imported into Unity and marked as interactable and grabbable. Additionally, a particle system (Unity 2018j) was added to the nozzle to emit particles imitating the oil pouring out of the can.

8.3.3 Service mode and rotor lock

Earlier, at the bottom of the wind turbine tower, there was a switch that stopped production and engaged the rotor lock. Usually, the technicians will engage a service mode first; then from the control cabinet in the nacelle, the rotor lock is engaged. In the Enercon turbine that is represented in this application, it is, in reality, both a rotor brake, and a rotor lock in the nacelle, but these are combined into a single switch for simplicity. Engaging the rotor lock will complete the task and stop the rotation of the rotor in the exterior scene.

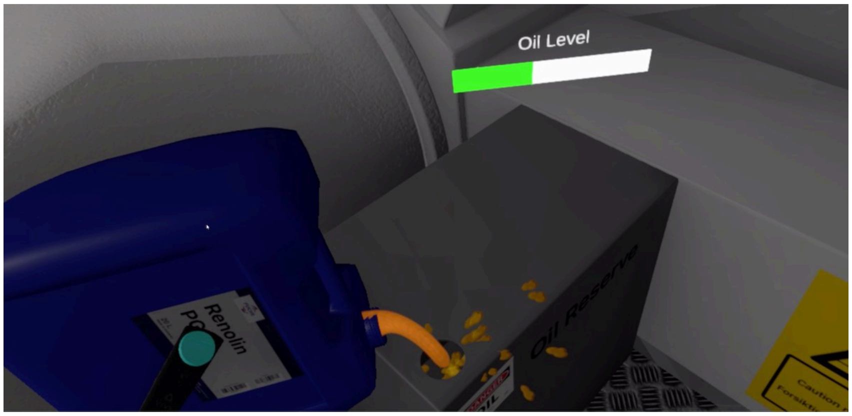


Figure 8.4: Filling oil into the oil reserve.

8.3.4 Inspecting cooling fans

In the control cabinet in the nacelle, cooling fans were inserted in order to implement a task where the user would have to inspect that all of them are functioning. Four fans were added, with a spinning animation created in Blender (which Unity can directly import and play). In Figure 8.5 the cabinet with the fans can be seen.

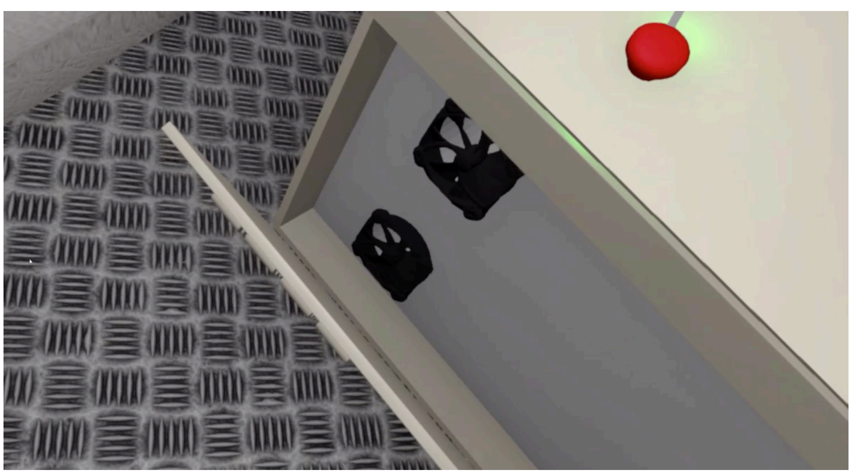


Figure 8.5: Inspecting the cooling fans in the low control cabinet in the nacelle.

8.3.5 Notifications

In the menu, the user can see what tasks are done, but there is no apparent feedback the moment they are completed in the application. In order to rectify this, a notification system was added, so when the user completes a task, a notification will pop up in front of the user along with a sound signaling the completion of the task as seen in Figure 8.6. The notification acts as immediate feedback to the user when they complete a certain task. After a brief pause, the notification will animate down and disappear.

Figure 8.6: A notification pops up in front of the user as they complete a task. The notification follows the HMD position, and will after a brief pause animate down to disappear.

8.4 Graphical Improvements

In this last iteration, more focus was put on the graphical quality of the application, with a new shader for the terrain texture, post-processing effects, more trees, grass, and simulated wind.

8.4.1 Environment improvements

The third iteration brought many improvements to the environment of the exterior scene, which is a requested feature. More rocks, trees, and grass were added along with a wind effect to make it look like trees and grass was waving in the wind. Both wind sound and wind turbine sound was added to make it all more immersive. Additionally, animated birds with bird sounds were added to bring the scene to life. In order to make the environment look less barren, a road was added to the island. Finally, a fog effect was added to both the terrain and the skybox to make it look more like an “infinite” horizon when standing on top of the wind turbine, making it less apparent that the terrain area suddenly stops. The scene with the improvements can be seen in Figure 8.7.

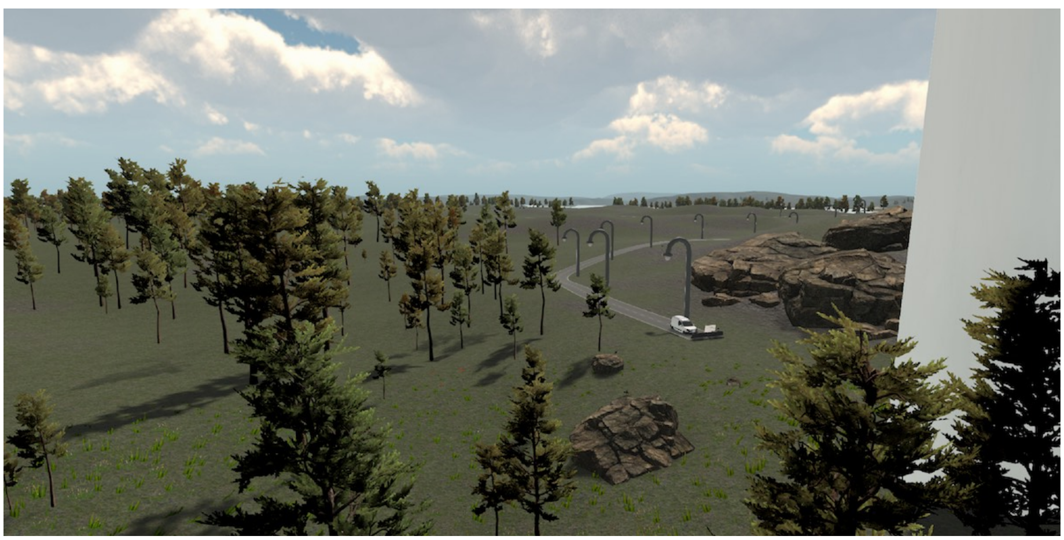


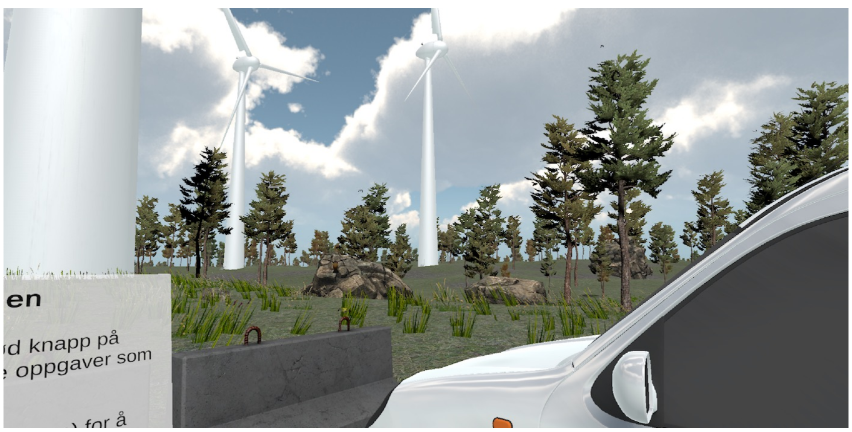
Figure 8.7: The exterior scene with added improvement, more rocks, trees, grass and fog.

8.4.2 Post-processing

Post-processing is a technique, or process, where visual effects and filters are applied to an image, before displaying it on screen (Unity 2018f). Post-processing can be used to reduce visual artifacts, add visual artifacts, or promote a special art style or mood in a game. Colors can be strengthened, weakened or altered and effects like film grain, motion blur or chromatic aberration can be applied to the rendered image Unity 2018f. In this application, the most relevant effects were deemed to be those that can increase the visual quality and decrease visual artifacts, therefore mainly two effects are applied: *color grading* and *anti-aliasing*. Applying these effects are done through using Unity's Post Processing Stack v2. Color grading is the process of altering the colors and luminance of the rendered image that is displayed to the user (Unity 2018c). In this application, color grading that made the final picture slightly warmer and more vibrant was applied, to increase the vividness of the nature and environment. Anti-aliasing is a term for post-processing algorithms that are designed to prevent (or alleviate) aliasing. In effect, this will reduce jagged edges on objects in the application but will introduce some level of blurriness as a side effect. The anti-aliasing algorithms available in Unity Post-Processing Stack v2 is Fast Approximate Anti-aliasing (FXAA), Subpixel Morphological Anti-aliasing (SMAA), Temporal Anti-aliasing (TAA). A summary of the different options are provided in Table 8.2. The one used in this application was chosen to be FXAA because of the performance and the ability to still deliver a decent level of quality, making it a compromise with decent quality and decent performance. The difference between no post-processing and all post-processing applied can be seen in Figure 8.8. It might be hard to dissect on these pictures, but edges in the lower picture are significantly less jaggy. It should be noted that the effect of anti-aliasing is more apparent when wearing an HMD and using the application.

Name	**Short**	**Description**
Fast Approximate Anti-aliasing	FXAA	The fastest and cheapest. Recommended for mobile applications.
Subpixel Morphological Anti-aliasing	SMAA	High quality. Not recommended on mobile, not supported on VR/AR.
Temporal Anti-aliasing	TAA	Advanced, more expensive. Recommended for desktop applications.

Table 8.2: The anti-aliasing algorithms provided by the Unity Post Processing Stack v2.



(a) The exterior scene without any post processing applied.

(b) The exterior scene with color grading and anti-aliasing applied.

Figure 8.8: Comparison between the exterior scene with and without post processing

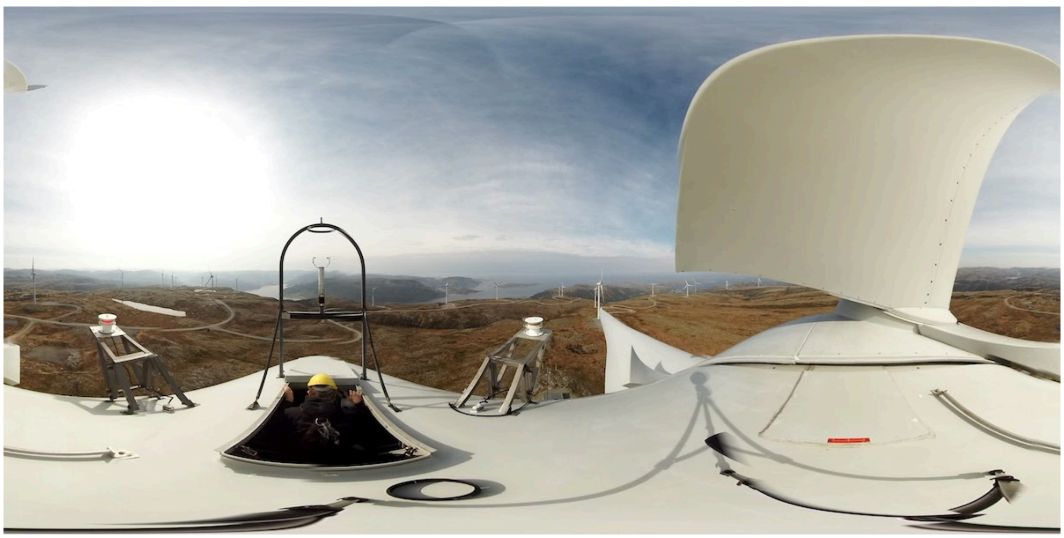
8.5 Score system

A score system is useful for the users to track their progress and how well they have done through the application. Since WindturbineVR has a different approach to the menu system than FiskeVR, the overview of the user's score is located in the game menu, in a separate tab. This tab can be seen in Figure 8.9. Each task has a corresponding score entry in the list, with the earned score at the right of the slash and the maximum earned score at the right. At the bottom, the total score is shown. The score earned for each task varies, and the end score will also vary depending on how the user completed the task. For example, if they climb the ladder without security fastened, they will earn half the amount of the maximum score.

Figure 8.9: The overview of the score of a user. As the progress through tasks users will earn points.

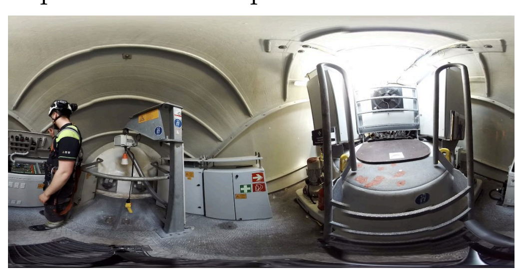
8.6 360-degree Videos

An element of FiskeVR is 360-degree videos that show parts of the workplace, captured on video and displayed in VR. TrønderEnergi orchestrated a trip to Bessakerfjellet Windpark in order to both meet wind turbine technicians and capture 360-degree videos from a turbine. The videos were captured with an Insta360 Pro camera with six lenses for a total of 8K resolution. Three videos were captured, one displaying the view from the top of a turbine out on the rest of the park (see Figure 8.10a), one from the nacelle of the turbine where the technician explains details about the control cabinet in the nacelle and how it controls the turbine (see Figure 8.10c), and finally a video from the bottom of the tower where the technician explains how the transformer cabinets work and the most frequent tasks performed there (see Figure 8.10b).



(a) Screenshot from the 360-degree video captured on the top of a wind turbine.

(b) Screenshot from the 360-degree video captured in the bottom of the tower of the wind turbine.



(c) Screenshot from the 360-degree video from the nacelle of the turbine.

Figure 8.10: The 360 videos captured at Bessaker Wind Park.

8.6.1 Importing 360-degree video to Unity

In order to display a 360-degree video in Unity, a *VideoPlayer* component can be used to render the video content, from either an imported asset file or an URL (Unity 2018k). For rendering the video around the user, a sphere object should be the target of the rendered video. To accomplish this, the VideoPlayer is set to target mode *Render Texture*; then a texture asset is set as the target for the rendering. This texture is then applied to the sphere. By default, all objects in Unity shows no backface, and the side of the face that is not in the same direction as the vertex normal is not rendered. This means that the engine *culls* the sides that are not facing in the vertex-normal direction. Therefore, without custom properties, the inside of the sphere will not be visible to the user. To change this, a custom shader with the command **Cull Front**, which will cull the front-facing faces, and not the back is useful. If no culling is wanted, **Cull None** can be used. By changing the culling, the inside of the sphere can be made visible, and the user can stand inside the sphere looking at the rendered 360 videos on the surface of the sphere.

8.7 The end product

This section briefly presents the application in its final state. If more detailed explanations or implementation details are available, each section here will refer to relevant sections in this iteration and earlier iterations that present implementation details.

8.7.1 Main Menu

The main menu scene did not change significantly throughout the development, as no major issues or needs were uncovered. It is a scene where the user is static, e.g., cannot move, and can choose between playing the tutorial or start the actual game. Selecting the options is done by the point pattern, described in Section 2.2.2. More detail about the functionality and implementation of the main menu can be found in Section 6.6.

8.7.2 Teleport tutorial

Apart from adjustments to reduce some errors, the teleport tutorial did not change significantly. It is a rather simple, but seemingly, an efficient tutorial where the user has to teleport from platform to platform to reach the goal. The platform the user should teleport to will glow green to indicate that it is the next platform to reach. Upon completion of the tutorial, the user is taken back to the main menu. The creation and more details about the teleport tutorial are explained in Section 6.7.

8.7.3 Exterior scene

The exterior scene is the largest scene in the application, and this is where the user spawns (see Figure 8.11). The user will spawn besides a car, in front of several wind turbines that all rotate. A huge green arrow pointing at one of the turbines indicates that this is the turbine that needs maintenance. The user is also greeted with a welcome message, telling them to open the menu to inspect tasks and read about how to do different tasks. The scene contains several details to make it more immersive: grass, trees, wind noise, wind turbine noise, flying birds, and bird noises. Additionally, the wind affects both the grass and the trees, so they wave in the wind. The exterior scene and implementation details are explained in Section 7.4 and 8.4.1.

Figure 8.11: The exterior scene of the application, seen from where the user will spawn.

The top of the turbine

At the top of the wind turbine the user will get a better look at the view and the entire exterior scene. They will look over the island Hitra, with Frøya at one side and the mainland of Trøndelag at the other. To make an effect of a an infinite horizon, a fog effect is added to both the terrain and the skybox used. The view from the top of the turbine can be seen in Figure 8.12.

At the top of the turbine (see Figure 8.13) there are two main things to do: change the lightbulb of the signal lamp and watch a 360-degree video from the top of a real-life turbine at Bessakerfjellet Windpark. One of the tasks in the application is to exchange the lightbulb in the signal lamp, and this is done by opening the protective hatch on the lamp, then take out the existing lightbulb by grabbing it with the controller. A new light bulb can be fetched in the menu under the gear-tab. When a new bulb is placed in the lamp, the task is

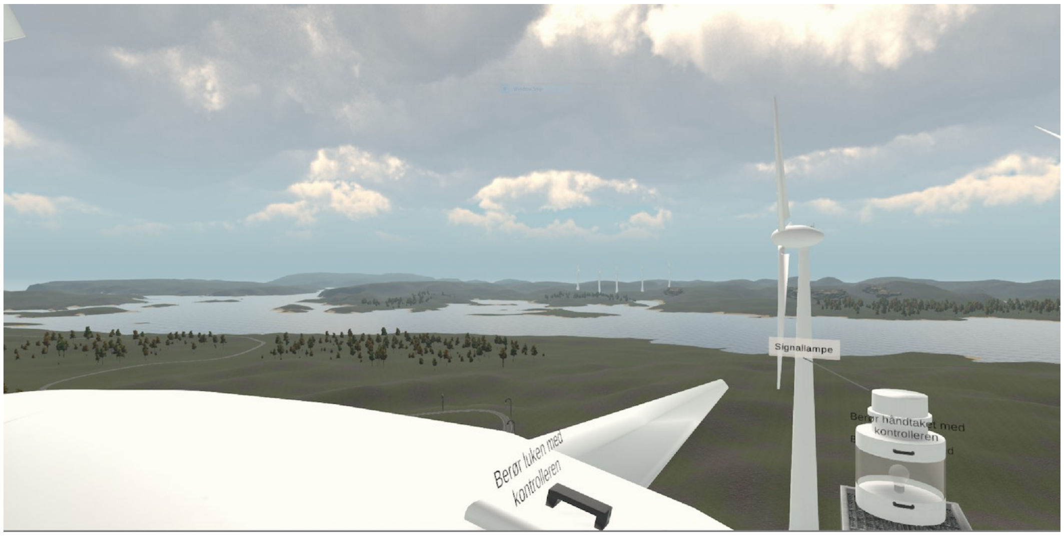


Figure 8.12: The view from the top of the wind turbine towards the mainland.

complete. The 360-video can be seen by touching the screen and pressing the trigger on the controller, and the user is instantly taken to the video.

Figure 8.13: The platform at the top of the wind turbine with the signal lamp to the left and the 360-degree video screen to the right.

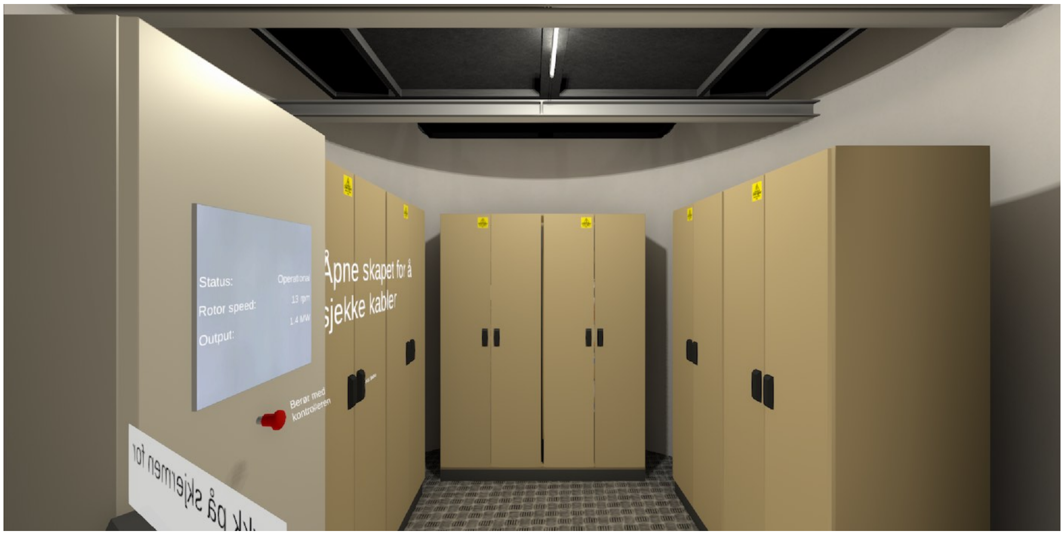


Figure 8.14: The view of the bottom of the wind turbine tower from the entrance. The button for turning the turbine into service mode can be seen at the left Large text on the cabinet to the left tells the user they should open the cabinets to inspect cables.

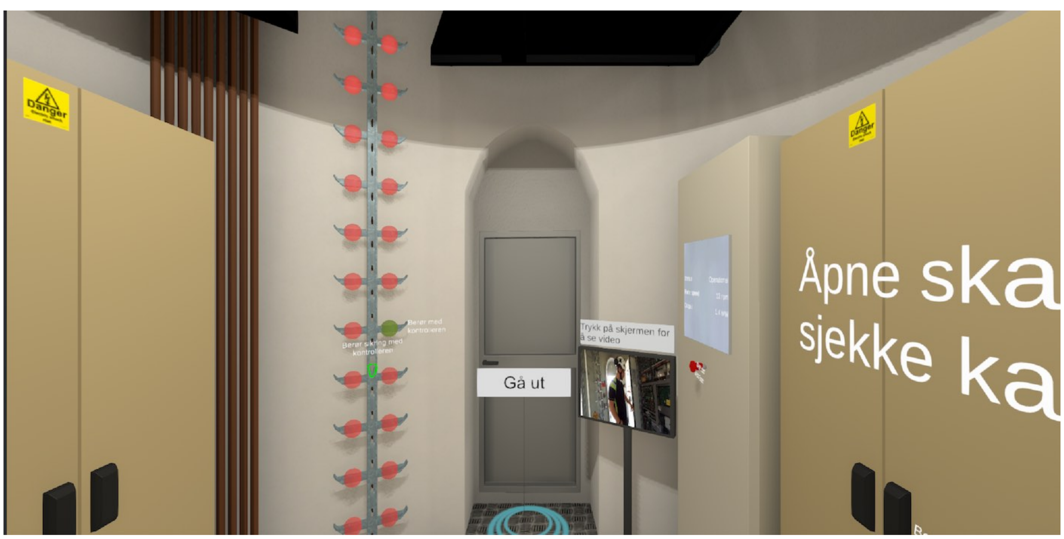


Figure 8.15: Standing inside the bottom of the tower looking towards the entrance. Here the ladder that needs to be climbed and the 360 video screen of the bottom tower is visible.

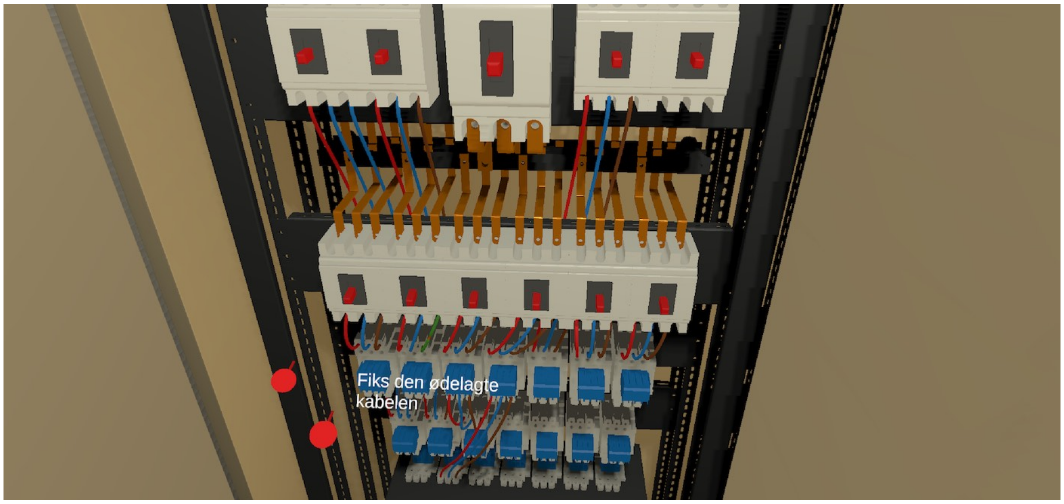


Figure 8.16: The internals of the transformation cabinets in the bottom of the wind turbine. The cables has to be inspected and fixed by the user.

8.7.4 Wind turbine tower

The tower is accessed by teleporting into a blue access point at the bottom of the stairs at the exterior of the turbine. The user will enter at the bottom (see Figure 8.14). This location houses three main tasks: switching the wind turbine into service mode, checking and fixing cables in the transformation cabinets and climbing the ladder to the very top of the tower. Switching the turbine into service mode is done by pushing the red button at the left cabinet seen in Figure 8.14. The climbing task is initiated at the ladder seen at the left side in Figure 8.15. To climb up the user has to sequentially grab the red spheres with the controllers, and drag themselves up, as they would in a real ladder. This will move the camera rig upwards and simulate climbing the tower. At the right side of the entrance in Figure 8.15 a video screen with a 360-video where a wind turbine technician explains what is inside the cabinets the user can see. Except for the control cabinet with the service mode switch, all cabinets can be opened by the user and inspected. The internals of the cabinet can be seen in Figure 8.16, where the user has to inspect cables and check if they are broken. Broken cables are identified by broken insulation on the cable, and for extra visibility sparkles and flashing light is added to the cable when broken. To fix the cable, the user touches the cable with the controller and press the trigger button. More detailed about implementation, creation, and functionality inside the turbine tower can be found in Section: 6.4, 7.2.2, and 7.3.

Figure 8.17: The view of the nacelle as the user reaches the scene. The two arrows in the middle are used for simple navigation down to the tower or up to the top of the turbine exterior.

8.7.5 Nacelle

When the user has climbed all the way up the tower, they will eventually reach the nacelle of the turbine. In the turbine there are three tasks to complete: turn on the rotor lock, inspect cooling fans and refill the oil reserve. In Figure 8.17 the starting point inside the nacelle is depicted. The two green arrows lead down back to the turbine tower and up to the top exterior, respectively. To use the arrow, the user have to touch them with the controller and press the trigger button. The rotor lock task can be completed by pushing the red button at the low control cabinet seen in Figure 8.18. Here the 360-video screen, where a wind turbine technician explains the functionality of the cabinet, is presented too. To inspect the cooling fans, the user will have to open the same control cabinet and verify that all the fans spin. The last task in the nacelle can be seen in Figure 8.19. Here the user has to pick up the blue oil can and pour oil into the oil reserve (the gray box to the right of the blue oil can).

Figure 8.18: The control cabinet at the left and the 360 video screen where a technician explains the functionality.

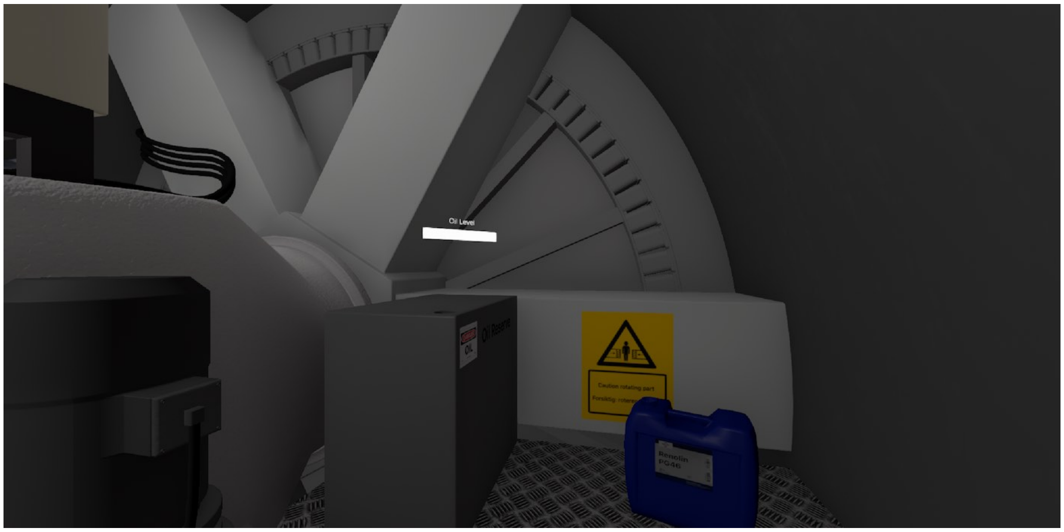


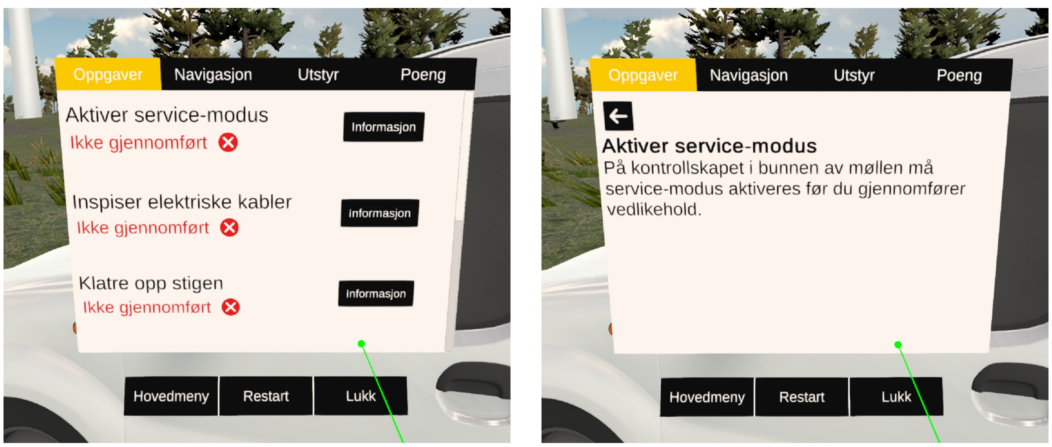
Figure 8.19: The oil fill task at the front right of the nacelle room.

8.7.6 In game menu

The in-game menu is made to be a versatile tool the user should return to both for navigation, detailed information on tasks, their current progress, score and retrieval of gear if needed. In reality, it is quite extensible, so it could be made to fit for any virtual internship that would need these features from a menu. The menu is opened with the menu button on the controller and will spawn in front of the user's HMD in world space at a comfortable distance. If an object is detected at a closer range, the menu will spawn closer to not collide with the object (up until a minimum distance). The user selects buttons and items on the menu with the pointing pattern. When the user hovers over a button, it will be highlighted in blue color, and haptic feedback in the controller will signal that the user is currently hovering over something clickable. When clicked, with the trigger button on the controller, the button will be highlighted in yellow to give feedback that the action was successful.

Tasks

The tasks in the application are presented in the *task* tab of the menu, as a list, as shown in Figure 8.20a. Each element has an indicator displaying whether the task is completed or not, and which will update when a task is completed. Also, a button on each element leads to a detail page that explains the task, why it is done, and hints at how and where to do it. This is presented in Figure 8.20b. The tasks are listed in the recommended order to complete them, but the user is free to do it in what order they want. When a task is completed, the user will receive a notification that pops up in front of their HMD, that lets them know what task changed status.



(a) The scrollable task list in the in-game menu.

(b) Detail page of a task, with additional info and hints.

Figure 8.20: The task tab of the in-game menu.

Navigation

The navigation tab displays a map of the turbine, where the user can select different parts of the turbine and immediately jump to that specific part. This acts as a faster way of exploring the different parts, instead of teleporting around, climbing ladders and entering doors. It is supposed to give the user a quick, effortless way to explore what they want.

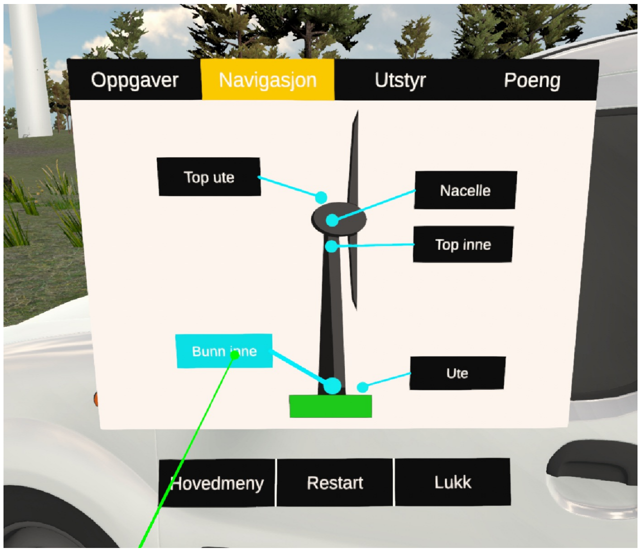


Figure 8.21: The navigation menu that the user can use to navigate around the application.

Gear

The gear tab is a list of the gear available in the application. For WindturbineVR the only gear placed there is a light bulb, but it could be more for the future development of this application or if other virtual internship applications utilize this menu system.

Score

The score tab displays the score the user has accumulated for each task, along with the maximum score that can be achieved. At the bottom of the list, the total score is shown. More details in Section 8.5.

8.8 User Testing

This section describes the final testing done on the application.

8.8.1 NAV Possibility Conference

At two occasions NAV's Possibility Conference was attended by the author. This conference is an internal NAV event where employees in NAV can get inspired and learn from employees from all over the country (Friess 2019). Some regions hold workshops at the event focusing on a specific topic or practice. NAV Trøndelag held a workshop with the topic "Can VR and game technology help youth?". The workshop was about the virtual internship project and the collaboration between NAV and NTNU, and how the usage of these applications can help NAV reach and guide youth in their choice of career or workplace. The general reception was good, and people are excited about the usage of technology in new ways with quotes like "This should be in all NAV offices in the country" and "This is a tool we have a use case for".

Some employees are inquiring a lot of the cost of the machines, HMDs, and development of such applications, which is understandable. There is also the fact of *maintenance* of such applications, fixing bugs and updating to be compatible with new versions of OS and other required components. Then there is the fact of teaching employees to use the hardware and the software in order to use it as a tool when working with the youth they are responsible for. These are all practical problems which indeed is solvable, but they are still essential to start thinking about as they are a hurdle to be able to actually test the applications as the tool they are made to be. Additionally, some employees comment on the fact that the developed applications are very specific to Trøndelag, especially WindtubineVR and FiskeVR, and that for a wider adoption in the country more widely relevant wokrplaces need to be developed.

Through all the testing that is done it is currently established that both NAV employees and youth are positive and eager to use these applications as tools.

The youth believes that they can experience new workplaces in a more immersive and informative way, but these are all preemptive tests. The next step should be deploying on a NAV office for actual use, and the employees seem both eager and ready for this.

8.8.2 Bessakerfjellet Wind Farm

The last phase of the project included an excursion trip to a wind park operated by Trønder Energi, where two wind turbine technicians tested the application. They did not answer questionnaires, but an informal discussion where conducted post-testing. In Figure 8.22, one of the technicians can be seen testing the application. Both technicians felt the application represented their workplace and typical day-to-day tasks on a turbine. Overall, both were positive and optimistic on the use of these applications as a a tool to present the job. Since the tasks, in general, are simplified in comparison to realistic electric and mechanical tasks that need to be done, they were pressed on this issue. Both said that albeit the simplifications, it still makes a realistic impression of the work and workplace. A quote from one of the technicians on the matter: "This absolutely gives an insight into how we work, even with simplified tasks", "To showcase the job, this is good", and "It is actually kind of scary to stand on the top". On lacks in the application, they felt the most pressing one is cooperation. As a wind turbine technician, you always work with a partner, and cooperation with that partner is an integral part of the job. This is, in fact, a necessary part of the virtual internship project in its entirety, as most workplaces naturally include elements of cooperation to some degree. They felt the number of tasks was sufficient: "You could have hundreds of different tasks, but to showcase the work this is enough". Too many tasks would possibly make the application to long and tiresome. They also commented on the fact that there was no need to winch up the necessary equipment to the top. In reality, they have to use the winch inside the turbine to carry heavy equipment or tools to the top of the turbine. The implementation of this whinch can be a point of future development. Additionally, there were some phrases

Figure 8.22: A wind turbine technician at Bessakerfjellet Wind Farm using the application.

used in the game that not necessarily coincided with what they used on the job, but the consensus here was that it was not of great importance. A feature request that has come up frequently in testing has been to measure time on tasks, especially the climbing. One of the technicians dismissed this and said that one of the most important things as a wind turbine technician was not to stress and always give a higher priority to security than time. This taps into the fact that it is crucial for these virtual internships that the *gamification* of the job does not impact the realism and realistic representation of the real job.

8.8.3 Evaluating at NAV Jobbhuset Falkenborg

The last evaluation testing was done at NAV Jobbhuset Falkenborg, where NAV employees themselves supervised most of the tests. This is the first test where developers or the researchers of the applications are not the ones supervising the tests. This can be a good way to both test out how the applications work when NAV employees use them with users, as well as reducing the effect of the researcher talking and interacting with the research subject, as mentioned in Section 3.4. It should be noted that for this test, a Windows MR headset made by HP was used, along with a laptop to run the application. Since this headset has a generally lower resolution and lower quality display, along with less accurate motion tracking compared to the HTC Vive the results can be impacted by this. Results from the questionnaire that were given to both NAV employees and NAV users can be seen in Table 8.3. Table 8.4 presents results from the questionnaire that only NAV users answered. Table 8.5 presents the result from the questionnaire that only NAV employees answered.

Table 8.3: Results from the questionnaire given to NAV employees and users that tested the application at the 28th and 29th of May at NAV Jobbhuset Falkenborg.

ID	**Question**	**Average Score**	**Median Score**
Q1	How will you evaluate the visual quality of the application?	3.91	4
Q2	How will you evaluate the quality of the interactions in the application?	4	4
Q3	I enjoyed using the application	4.73	5
Q4	The tasks I were given was simple to complete	4.64	5
Q5	There was a lot that seemed illogical and unpredictable	1.82	2
Q6	I felt motion sick	1.82	2
Q7	The tasks I were given seemed realistic	3.82	4

Continuation of Table 8.3			
ID	**Question**	**Average Score**	**Median Score**
Q8	I was in need of instruction when using the application	3	3
Q9	After completion I had a better understanding of what the job is like	3.91	4
Q10	Time limits on tasks would have made the application more interesting	3	3
Q11	Timelimits to solve tasks would make the application more interesting	2.91	4
Q12	A possibility for cooperation would have made the application more interesting	3.55	4

Table 8.4: Results from the questionnaire given to NAV users that tested the application at the 28th and 29th of May at NAV Jobbhuset Falkenborg.

ID	**Question**	**Average Score**	**Median Score**
Q1	I would like to use such applications often at NAV/school	4.86	5
Q2	Such applications can make me more confident when applying for a job	4	4
Q3	Such applications can give me a better understanding of the workplace	4.43	4
Q4	Such applications can give me more confidence in execution of workplace tasks	4	4

Continuation of Table 8.4			
ID	**Question**	**Average Score**	**Median Score**
Q5	Such applications should be a part of NAVs service to job seekers	4.71	5
Q6	Such applications should be a part of career guidance in schools	5	5
Q7	The application was easy to use	4.57	5
Q8	I think I would need help from a technical skilled person to use the application	2.86	3
Q9	I think the different parts of the application matched together	4	4
Q10	Such applications can give me increased motivation to apply for jobs	3.86	4
Q11	I will feel safer in a new job if I have trued such applications prior to starting	4.14	4
Q12	Such applications give me a feeling of *presence* in a workplace	4	4
Q13	I think it was easy to immerse in the application	4.4	4

Table 8.5: Results from the questionnaire given to NAV employees that tested the application at the 28th and 29th of May at NAV Jobbhuset Falkenborg.

ID	**Question**	**Average Score**	**Median Score**
Q1	I want to use such applications in my work with users.	4	5

Continuation of Table 8.5			
ID	**Question**	**Average Score**	**Median Score**
Q2	Such applications can be useful to prepare job seekers for an interview.	5	5
Q3	Such applications can introduce the user for workplaces and typical tasks.	5	5
Q4	Such applications can motivate users to apply for jobs.	4.75	5
Q5	Such applications can give the users a feeling of mastery in the process of applying for jobs.	4.75	5
Q6	Such applications should be a part of NAVs service to job seekers.	4.75	5
Q7	Such applications should be a part of career guidance at schools.	4.75	5
Q8	A larger focus on game elements would make the applications more interesting	3.25	3
Q9	The application was easy to use	4.25	4.5
Q10	I think I would need help from a technical skilled person to use these applications as a part of NAVs service.	2.5	2.5
Q11	It would be easy to integrate such applications as a part of NAVs service.	4.5	4.5

8.9 Performance Test

During development of the application a lot of thought and time has gone into optimizing models and effects in order to keep the performance at an acceptable level throughout a play-through. It is important to not have performance as an afterthought, but keep it in mind along both the design and implementation of an application because identifying the aspects of an application that results in worse performance can be hard. To exemplify how this focus on performance along the development process can result in a better level of performance a framerate test was done on both FiskeVR and WindturbineVR. The results can be seen in Table 8.6.

Application	**Average FPS**
FiskeVR	46.590
WindturbineVR	86.963

Table 8.6: The average framerate measured in both FiskeVR and WindTurbineVR.

The results clearly shows that WindturbineVR has a significantly higher Frames Per Second (FPS) than FiskeVR, where FPS refers to the number of frames that the GPU render each seconds. The tests were performed on a high end desktop machine with and i9 7900X CPU, GTX 1080 GPU and 32GB of RAM. The connected HMD is a HTC Vive Pro. The data was captured with Fraps, starting the benchmarking at the main menu of both applications and stopping after a quick play-through of about 5 minutes. The accuracy of the test is not perfect, as the stopping and starting of the benchmark is done manually and the play-through is also manual, so it is no rigid predetermined ”route” that can be repeated to verify the results properly. However, it does display just how important it is, and how much of an impact it can have when performance is a focus throughout the development of a game. It is also important to note that this test was performed at a powerful desktop system, while in real life usage NAV will most likely employ laptops for using these application, making the performance of the application even more important.

Chapter 9

Discussion

First and foremost, the main outcome of this thesis is the result of the Design and Creation process, which in this case outputs an *instantiation*; WindturbineVR. In Section 9.2, results from the evaluation tests for FiskeVR and WindturbineVR are compared and discussed. This thesis has described and evaluated the current FiskeVR application in the virtual internship project, in order to find pressing points on what works and what can be further improved and experimented with in the project. The knowledge gained from the literature review and the case study on FiskeVR has been used to form, design, and develop WindturbineVR, which is a new workplace and a new virtual internship application. The value of these virtual internships is discussed in Section 9.3. Development of the new application uncovered several important points and techniques for transforming a workplace into a virtual version. This process is discussed further in Section 9.4. During the development, guiding, tutorials and feedback were focused on as an important point, to understand if this has any impact on the user-friendliness and perceived usefulness of the application. No significant increase in utility is found, but it is still an important feature and does increase the perceived *ease of use* of the application. This aspect is discussed in Section 9.5. The development itself needs a foundation to build upon, which in FiskeVR is the SteamVR SDK. In WindturbineVR it was experimented with using a platform agnostic SDK, VRTK. The usage of this and how it accelerated development is discussed further in Section 9.6.

9.1 Limitations

This project has been limited in time and resources, both for development and research. The initial case study was performed during the fall of 2018, while the design and development of WindturbineVR were done in the spring of 2019. As a single developer (the author) created all models, art, design, and code for the application, there is a limited amount of features that are possible to implement, meaning prioritization needs to be done. Additionally, there is a minimal amount of resources to be used to buy assets and software from external providers, and the ones that were bought were with personal resources.

The developer had no extensive experience with modeling, game development, or VR development, meaning that during the process, a lot needs to be learned taking time from the actual development of new features, and validation of new concepts. The information and data about the workplace rely on information from TrønderEnergi, and access to technicians and correct personnel is highly dependent on their availability. The users that test the application is highly dependent on what users NAV can supply; the researcher can not willingly pick them. The age group, gender distribution, distribution between employees, and the job seekers current position in the process towards a job cannot be controlled, as it merely depends on who is available at the time.

When comparing the test results between FiskeVR and WindturbineVR, it is important to stress that these applications differ not only in techniques and implementations details, but the nature of the tasks are quite different. On the other hand, if they did not have differences, comparing them would be of no use. It is important to stress that the comparisons in Section 9.2 cannot be seen as statistical evidence, and bear no statistical significance because of both the difference in nature of the tests that were performed and the applications that are tested. The discussion that comes will discuss indications and trends across the results. Knowledge from literature and related work, the numerous tests the author has supervised and development experience are used to interpret the quantitative results. Additionally, the tests are performed on two different hardware systems; FiskeVR is tested on a desktop with a HTC Vive connected, while WindturbineVR was tested with a laptop and a WindowsMR HMD. By experience, WindowsMR systems have poorer position tracking of both controllers and the HMD, which can induce more break-in-presence moments, as described in Section 2.1.3.

The reader is also encouraged to read the evaluation section in the method chapter, Section 3.4.

9.2 Comparing evaluation results

This section will present a comparison and discussion of the evaluation results for FiskeVR and WindturbineVR. Comparing of results is done in order to identify areas where the two applications seem similar, and where they differ. The results from FiskeVR is from the evaluation test that was held at the IMTEL VR Lab at Dragvoll at the 30th of October 2018 (presented in Section 4.2.3). The results for WindturbineVR is from the evaluation test held at NAV Falkenborg at the 28th and 29th of May 2019 (presented in Section 8.8.3).

9.2.1 Common questionnaire

Table 9.1 presents results from the evaluation tests where both job seekers and NAV employees answered, comparing FiskeVR with WindturbineVR. In Figure 9.1 the average results are drawn in a graph, which shows that, in general, users answer very similar between the applications.

Table 9.1: Comparing results of the questionnaire given to all users, from the NAV evaluation of Fiske VR at the 30th of October 2018 and the WindturbineVR evaluation at the 28th and 29th of May 2019.

ID	**Question**	**FiskeVR**		**WindturbineVR**	
-	-	Average	Mean	Average	Mean
Q1	How will you evaluate the visual quality of the application?	3.7	4	3.91	4
Q2	How will you evaluate the quality of interactions in the application?	3.77	4	4	4
Q3	I enjoyed using the application.	4.13	4	4.73	5
Q4	The task I were given was easy to complete	4	4	4.64	5
Q5	There was a lot in the application that seemed illogical or unpredictable.	2.04	2	1.82	2
Q6	I felt motion sick	2.27	1	1.82	2
Q7	The tasks I were given seemed realistic	4.19	4	3.82	4
Q8	After completion I had a better understanding of what the job is like	4.09	4	3.91	4

WindturbineVR scored slightly better in graphical quality than FiskeVR, but not significantly with average scores of 3.91 and 3.7 respectively. It should be noted that FiskeVR was tested on desktop machines with HTC Vive Pro or Original HMDs connected, and WindturbineVR was tested on a laptop with an HP Windows MR HMD. The HP headset provides lower resolution, lower quality display, smaller FOV, and more mediocre motion tracking than the HTC Vive systems. Therefore, it seems the focus on graphical quality and

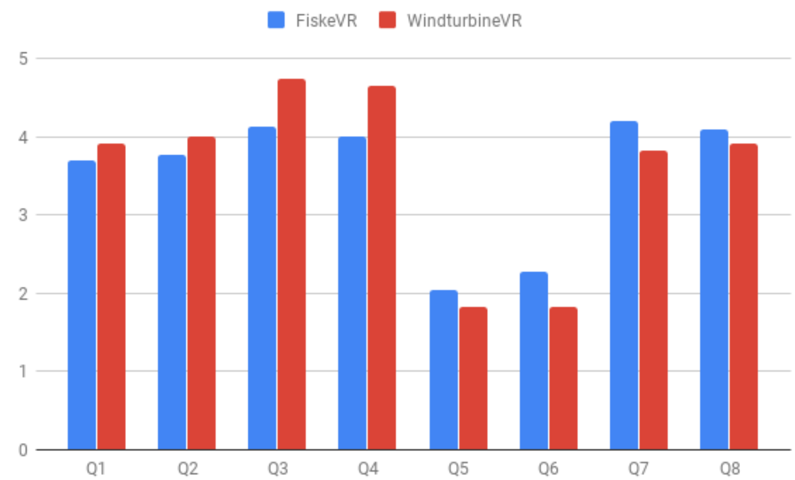


Figure 9.1: Graph comparing the average results given in Table 9.1.

post-processing has paid off to some degree, as the results show that WindturbineVR still kept pace with FiskeVR, even surpassing it slightly. Immersiveness is essential to induce presence, and a continued focus on a balance of performance and graphics is essential since one of the key characteristics of a VR application is the requirement of real-time performance while maintaining presence and realism, as described in Section 2.7.1. Additionally, graphics and audio are vital to induce sensory curiosity in the user, which can motivate the user to explore and use these applications (Section 2.5.2).

WindturbineVR scored higher on **Q4**: "The tasks I were given was easy to complete", with a score of 4.64, versus 4 for FiskeVR. It could be that the increased focus on guiding and usability contributed to the fact that the users felt it was easier to complete the tasks. Also, another contributor could be the nature of the tasks themselves, where WindturbineVR has more procedural and descriptive tasks; FiskeVR has more interactive and "free" tasks. It is not inherently positive that the users deem the tasks easy to complete; it could be that this means that the tasks are less challenging and engaging for the user, and thus less enjoyable. However, WindturbineVR did score higher than FiskeVR in **Q3**: "I enjoyed using the application", which counts against this conclusion. FiskeVR scored higher for both **Q7**: "The tasks I were given seemed realistic" and **Q8**: "After completion I had a better understanding of what the job is like". This most likely leads back to the difference in simplification of the tasks. The tasks in WindturbineVR is simplified in a higher degree than in FiskeVR, which then leads to them seem less real to the users. An impression of less realistic tasks can mean a feeling of not understanding the job at the same level. It can seem like realism has been traded, in a too high degree, for user-friendliness and simplicity in WindturbineVR. Future virtual internships should simplify the tasks to a lesser degree than what is done in WindturbineVR, in order to increase the realism.

Regarding motion sickness, both applications have a decently low score (meaning the users did not feel motion sick). FiskeVR has an average of 2.27 and WindturbineVR 1.82. FiskeVR's higher score can be because of a few extremes, since the median is very low, at just 1. The leading cause of motion sickness in WindturbineVR is, by experience and observation, if the user falls down the ladder or the turbine. In FiskeVR, the leading cause is usually driving the boat. Both these cases are *vection*, which is an illusion of self-motion. Vection is found to induce motion sickness in some users (see Section 2.1.4). For both applications, there is a possibility to skip these parts, if the user feels nauseous. Future applications should not refrain from implementing parts of the workplace that includes interactions that can induce motion sickness in some users; then it is better to keep an option to skip these parts and let the users that can and will, try them out.

9.2.2 Job seeker specific questionnaire

Table 9.2 present a comparison between the results of the questionnaire given exclusively to job seekers that tested the application. Figure 9.2 presents a graph that displays the average results of each question. As for the common questionnaire (Section 9.2.1), the answers for both applications are relatively similar.

Table 9.2: Comparing results from the quesionnaire given to job seekers from the NAV evaluation of Fiske VR at the 30th of October 2018 and the WindturbineVR evaluation at the 28th and 29th of May 2019.

ID	**Question**	**FiskeVR**		**WindturbineVR**	
-	-	Average	Mean	Average	Mean
Q1	I would like to use such applications frequently through NAV/school.	4.31	4.5	4.86	5
Q2	This type of application can make me more confident when applying for a job	4	4	4	4
Q3	This type of application can give me a better understanding of the workplace	4.38	4	4.43	4
Q4	This type of application can give me more confidence in execution of workplace tasks	3.88	4	4	4
Q5	This type of application should be a part of NAV's service to job seekers	4.44	5	4.71	5
Q6	This type of application should be a part of career guidance in schools	4.67	5	5	5
Q7	The application was easy to use	4.19	4	4.57	5
Q8	I would need assistance from a technical skilled person to start using the application	2.94	3	2.86	3

Continuation of Table 9.2					
ID	**Question**	**FiskeVR**		**WindturbineVR**	
-	-	Average	Mean	Average	Mean
Q9	This type of application can gie me increased motivation to apply for jobs	4.13	4	3.86	4
Q10	I would feel safer in a new job if I had tried this type of application prior to starting.	4	4	4.14	4
Q11	This type of application give me a feeling of presence in the workplace.	3.5	4	4	4
Q12	I think it was easy to feel immersed in the application.	4.38	5	4	5

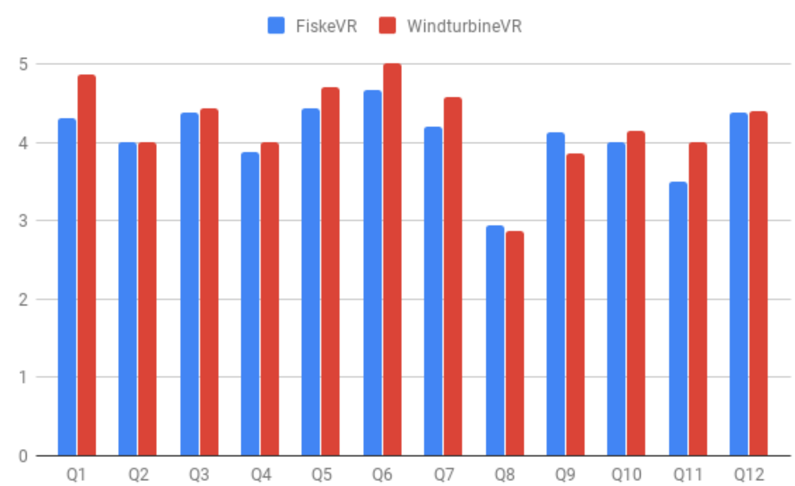


Figure 9.2: Graph comparing the average results given in Table 9.2.

The results from the job seeker specific questionnaire show that the users feel both FiskeVR and WindturbineVR should be available as a service from NAV and as a part of the career guidance in schools. With average results well above 4, and closing in on 5 for both applications for the questions **Q1**: "I would like to use such applications frequently through NAV/school", **Q5**: "This type of application should be a part of NAV's service to job seekers" and **Q6**: "This type of application should be a part of career guidance in schools.". For all these three questions WindturbineVR seems to receive a slightly higher score, but it does not seem to be a significant difference, and the point is that the users believe these applications, in general, is useful and should be used for career guidance. This point is essential; the users do believe that the concept has merit, and they feel and believe schools and NAV should use it. An example citation from a user: "You get to actually experience how it is to work there, instead of just reading it from a book or watching some video". The users themselves believe that the applications are valuable, and that is important. Further, it will be essential to establish whether the applications have real utility when used in the field, or if the excitement stems from the fact that these applications and the new technology are just exciting and fun for the youth, and not useful as tools for career guidance. If it turns out that virtual internships are the future of career guidance, then it is excellent that the job seekers are both excited and eager to use them.

Statement **Q7**: "The application was easy to use" got an average result of 4.19 (FiskeVR) and 4.57 (WindturbineVR). The increased focus on guiding and information in the application seems to make the user feel that it is easier to use, and the median score for this statement is 4 for FiskeVR and 5 for WindturbineVR. By spreading the information out in the world, making it available when the user needs it, applications can be more interesting and engaging as well. This is one of the strengths of games, as uncovered in Section 2.5.2.

Two conflicting results are the answers on **Q11**: "This type of application gives me a feeling of presence in the workplace" and **Q12**: "I think it was easy to feel immersed in the application". FiskeVR got a lower average score on immersion when compared to WindturbineVR (3.5 vs. 4), but a higher average score on presence (4.38 vs. 4). The reason is unknown, and it could be coincidences, along with the interpretation of the questions in Norwegian. Further speculation on the reason for the conflicting result will not be performed. Nevertheless, it seems that graphical quality and audio should be a future focus, as this will increase the chance of inducing presence, and increase the degree of experienced presence.

9.2.3 Employee specific questionnaire

Table 9.3 shows the comparison of results from the questionnaire given to NAV employees only after testing the application. Figure 9.3 displays a graph where the average result of each question is plotted.

Table 9.3: Comparing results from the questionniare given to NAV employees from the NAV evaluation of Fiske VR at the 30th of October 2018 and the WindturbineVR evaluation at the 28th and 29th of May 2019.

ID	**Question**	**FiskeVR**		**WindturbineVR**	
-	-	Average	Mean	Average	Mean
Q1	I want to use this type of application in my work with users	4.8	5	4	5
Q2	This type of application can be useful to prepare job seekers for interviews	5	5	5	5
Q3	This type of application can introduce users for workplaces and typical tasks.	5	5	5	5
Q4	This type of application can motivate users to apply for jobs	4.6	5	4.75	5

Continuation of Table 9.3					
ID	**Question**	**FiskeVR**		**WindturbineVR**	
-	-	Average	Mean	Average	Mean
Q5	This type of application can give users a feeling of mastery in the job seeking process	4.2	4	4.75	5
Q6	This type of application should be a part of NAV's service to job seekers	4.8	5	4.75	5
Q7	This type of application should be a part of career guidance at schools	4.8	5	4.75	5
Q8	The application was easy to use	4	4	4.25	4.5
Q9	I would need assistance from a technical knowledgeable person to start using this type of application as a part of NAV's service.	3.5	4	2.5	2.5
Q10	It would be easy to integrate such applications as a part of NAV's service.	3.75	3.5	4.5	4.5

The Figure in 9.3 shows that the answers between the applications are, again, similar to each other. Employees at NAV are positive to the usage of these applications in their work with users and feel that it can inform (Q3), motivate (Q4) and induce a sense of mastery (Q5) in the users. Additionally, they are favorable to the usage of these applications as a part of NAV's service (Q6) and as a part of career guidance at schools (Q7).

For employees, WindturbineVR also scored slightly higher on usability com-

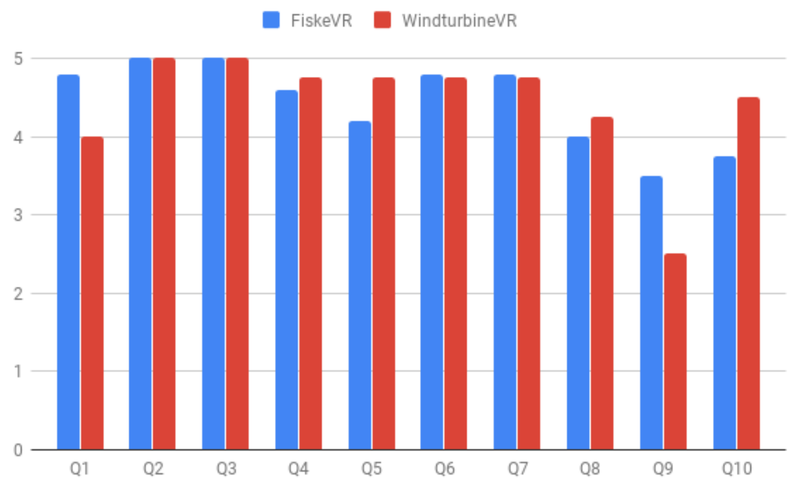


Figure 9.3: Graph comparing the average results given in Table 9.3.

pared to FiskeVR, with an average score of 4 for FiskeVR and 4.25 for WindturbineVR on **Q8**: "The application was easy to use". Even though it is not a massive difference, it reinforces the trend that the users appreciate the increased focus on guidance, information, and feedback.

The most significant difference on this questionnaire is rather peculiar; **Q9**: "I would need assistance from a technically knowledgeable person to start using this type of application as a part of NAV's service" received an average of 3.5 for FiskeVR and 2.5 for WindturbineVR (lower is better), even though it is not inherently any more challenging to run FiskeVR than WindturbineVR. The same effect can be seen on **Q10**: "It would be easy to integrate such applications as a part of NAV's service", where FiskeVR score 3.75 and WindturbineVR 4.5 (higher is better). One of the most significant differences of the tests is that FiskeVR was supervised by the author and two other NTNU students, with proficiency in VR, at the IMTEL VR Lab, while the WindturbineVR test was supervised by NAV employees, at a NAV building. It could be that the employees, seeing other colleagues handling the gear and supervising the tests, gained more belief in the fact that they could do the same thing.

This could mean that by deploying these applications at NAV offices, employees would gradually be more positive and feel better equipped at using the technology as they saw other colleagues use the gear. Of course, this implies that some employees initially utilizes the applications.

9.3 The value of a virtual internship

This discussion relates to **RQ1**: “What is the current value of virtual internships?”

The fact that a virtual internship has merits towards conveying realistic and useful information about a workplace is through results and discussion, presented in Section 9.2, fairly established. Both students, pupils, NAV users, and NAV employees see this as a valuable tool in career guidance and for informing about a job. It can be a valuable tool towards taking a choice, but for real-world usage, more workplaces are needed. A virtual internship is a form of serious game, with the primary learning outcome being *affective*; it affects the users’ internal state and can influence choices, actions, and motivation. A second important learning outcome is declarative knowledge about the workplace and the tasks. Besides, additional learning outcomes can, and will most likely, exist, but these may vary depending on the simulated workplace. A virtual internship should supply clear goals, tasks that might require some level of skill, and feedback on said tasks. These points enable the users to immerse in the experience, along with high fidelity graphics and audio. Throughout the testing, in both FiskeVR and WindturbineVR, the feedback points to the fact that both employees, the industry and the users think the potential for these applications to have an impact on how we do career guidance and recruitment is present.

By simulating a workplace in VR, with relevant tasks, a wide variety of jobs and careers can be depicted. Some of the strengths in workplace simulation for training purposes are found to be the fact that they are cost-effective, flexible for change and that they can simulate a wide variety of scenarios.

All these advantages is possibly present for virtual internships as well. When first developed, they can be a cost-effective alternative to short term real-life internships and company visits. They can be flexible and modified to fit the current status of the represented industry or workplace. A significant advantage is the possibility to simulate a wide variety of scenarios, that might be unavailable to most job seekers. They can experience and understand jobs they have not even thought about as options.

An important point is that testing done so far has been quite clinical or artificial. The fact that users like the concept and believe it has real advantages is a huge step towards establishing the actual utility, but there is one big part that still needs to be researched. What benefits and utility do these applications provide in actual usage by NAV? If these applications are going to be used in the real world, they need to show results in order to be worth developing. The current results of the project are based on what the users and employees of NAV believe, not evidence of actual utility in real-life usage. Will the usage of these applications by NAV, as tools, help people find a job quicker? Will it make the career guidance made by NAV employees have higher quality? Will it reduce the number of people that drop out of school or quit their job because of a wrongly chosen career? These are important questions and have to be further researched. Further tests where NAV employees use these as actual tools should then be performed, and the users' impressions when they are used as tools should be gathered, along with studying longer-term implications.

Another point is that both FiskeVR and WindturbineVR does not represent broadly relevant workplaces nationwide, as mentioned by NAV employees from other parts of the country (see Section 8.8.1). In order for these applications to be utilized in NAV offices across Norway, more workplaces that are relevant in more areas need to be developed. Some examples here are carpenter, shop employee, and hairdresser. By implementing workplaces that are more relevant in more places, the adoption of the usage could be better, and more data on the results of the usage could be gathered.

9.4 The process

This discussion relates to **RQ2:** “What tools and processes are useful and necessary for creating a virtual internship and representing a workplace realistically?”

For each component, part, scene or a specific section of the workplace, in this case, the wind turbine, there is a set of steps that are needed to take it from reality to virtuality. This process of bringing the physical to the digital space can be time-consuming, but with the right data and consistent workflow, it is quite feasible. It is important that the workplace is realistically represented; this is stated in both evaluation reports made for the project earlier. In order to enable this a more rigid methodology for development should be made (Prasolova-Førland, Fominykh, and Ekelund 2019). Through development it was found that the main steps in the process are (as seen in Figure 9.4) gathering *data*, creating a *digital representation* (3D model), applying *visual effects*, adding *interactivity*, and finally including the component into the application, applying necessary logic and code to include it in the *tasks* of the workplace. The process from reality should take into consideration real-time performance, realism, presence, and the physical properties, function, and behavior of the objects or components, as seen in Section 2.7.1. Presence cannot be guaranteed in any way, but by making the components in high fidelity, and close to the real-life counterpart, and adding believable visuals and audio, it can aid the user into experiencing some degree of presence.

Important to note here is that the process in this figure is supposed to depict the steps for one component, or ”part” of the workplace. It is not meant to describe the whole process of development, because that should not be a linear process, but an iterative one with user involvement as used in this thesis (see Section 3.1.2). The process for a single component can be seen as an incremental process, while an agile project as a whole should be both incremental and iterative (see Section 2.7.1). This way, many problem areas

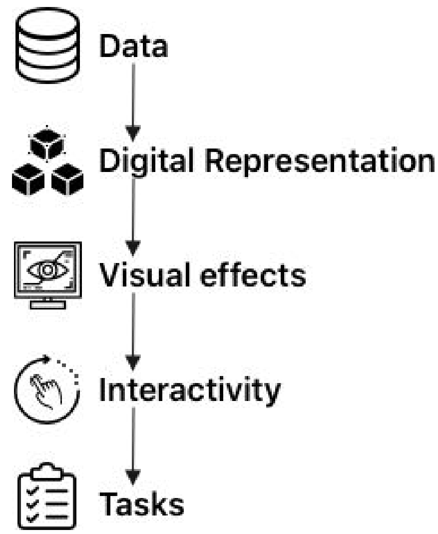


Figure 9.4: The general process of transforming data to a complete component in a virtual internship. The process takes information about a component, then transforms and incorporates it into the virtual workplace.

can be identified and eradicated under development. Additionally, the process is not meant to be a strict one-way route as the figure might imply. Sometimes, for example, when applying interactivity, one can realize that part of the model needs remodeling in order to support some specific functionality properly, and parts of the process need to be revisited.

In the first step of the process, data about the construction of the work environment needs to be gathered along with information about relevant tasks and systems that needs to be created. What properties does the component that is to be represented digitally possess? Data about looks, material, function, and dimensions should be gathered. This data can be gathered from various sources: pictures, videos, textual descriptions, talking to knowledgeable personnel, and so on. In this thesis, both research on the internet, information from employees at TrønderEnergi, and direct digital representation in the form of a 3D scan was used as the data foundation. An understanding of the workplace needs to be formed, along with more tangible representations of the component that is to be recreated. Since the various components will vary in both detail, importance, and nature, a detailed process for the data generation can be hard to create, and maybe even futile. The data generation will most likely vary from workplace to workplace, and also be related to resources available for the project and what resource companies in the industry can provide. Specifically, this means that what data to collect, and how to collect it largely depends on what workplace that is to be represented. The correctness of the collected data impacts the final correctness of the application; this is why contact with the industry in this phase is so important. Correctness, here, means that the final application represents the workplace in such a way that the users get an impression of how it is to work in that specific line of work.

When done collecting data, the component needs to be visualized in 3D space. The optimal process would be automated so that it could be fed with specific data, and output the 3D model of the component. It was experimented in using 3D models generated directly from point clouds of a wind turbine, but this showed up futile as the resulting models are both heavy to render and lack detail. Separating components from the model can be hard, and interactivity can be hard to add to these uniform models. The technique that is deemed necessary, if no finished models are available, is manual work to create 3D models of the relevant components to represent the workplace. When creating these models, it is essential to keep performance in mind and use no more triangles than necessary to provide a satisfactory result. In this thesis, Blender was used as the modeling tool for all objects and parts in the wind turbine, both interior and exterior; and since Blender is solid, free and open source it is preferred. Unity offers directly importing Blender files into the project so that it can be easily changed in Blender later without having to keep a separate Blender project where files can be exported from.

Along with constructing the model, low detailed versions (low poly) should be created where relevant, to support varying level of detail in the game engine (Section 7.3). Providing these low poly versions can improve and keep performance in check, which is even more important in VR than for regular desktop games. Having well-performing models can enable a higher level of graphics and detail while still maintaining the frame rate that is needed for a comfortable experience in VR. Presence is enabled by immersive high definition graphics, in addition to the user's context. It is therefore vital along the development process to continuously weigh performance and graphics. The advantages of keeping performance in mind during development, especially through the use of LOD, is evident in the performance test performed in Section 8.6.

After manually creating the model, or acquiring it in some other fashion, other visual aspects need to be created or acquired: textures, normal maps, lighting, and graphical effects. These can be downloaded where available or created with inspiration from the data collected about the modeled component. Adding such details can be a tiresome process, and for a solid result, professional artists should be used (which is also the case for the modeling of the component). The models should be appropriately UV-mapped to be able to apply textures, normal maps, lighting, and other details to the model for increased fidelity and realism. For the application as a whole, post-processing effects should be applied to increase the visual fidelity and quality. Especially, anti-aliasing is a useful tool in VR to decrease the effect of jagged edges on objects in the environment. In general, the visual step can take the realism of the application one step further.

When the visual aspects are in order, interactivity can be applied: relevant components like physics and colliders can be applied, and the objects can be included in the applications logical workings. Around all these objects or components, the tasks of the workplace have to be designed and implemented, including systems for tracking completion, score, etc.

9.5 Focusing on tutorials, guiding and feedback

This discussion relates to **RQ3:** "How do more focus on information, in-game guiding, and direct feedback impact the perceived usefulness and user-friendliness of a virtual internship?"

Compared to FiskeVR, Windturbine VR includes more guiding and hints along the way for users to follow, with textual descriptions for objects that indicate how to interact. This is an area which taps into the usage of game elements in the application and is in the introduction found to be an area where further experimentation is needed (Øygardslia et al. 2018). It is vital to take inspiration from game design literature and game technology. As explained in the literature, it is found that games have a unique way to spread information and apply it to the world that it is relevant for. Users should not be overwhelmed with information, all at once, but gradually be introduced to concepts when needed.

More guides, information, feedback, and tutorials do seem to make the user more independent when using the application, in regards to the amount, and detail of the instructions they need in order to be able to use and complete the tasks in the application. The results show decently low scores on the statement "I was in need of instruction when using the application", with a final average and median score of 3 (reported in the evaluation test in Section 8.8.3). Also, as shown in the comparison in Section 9.2, the application has a high score for statements that relate to the ease of usage compared to FiskeVR. It is evident when testing with users that less explanation is needed than for FiskeVR. The notification system does seem to make the user more aware of when a task is complete, making them more independent.

What the increased focus on guiding does not do, is impact the perceived usefulness of the application by the users, as there is no clear evidence that the user rate the usefulness significantly higher for WindturbineVR than for FiskeVR. On the contrary, Windturbine VR received a slightly lower score for the statement “After completion I had a better understanding of what the workplace is like” than FiskeVR. This could be numerous reasons, but one of the most obvious ones could be the nature of the tasks and the implementation of them in WindturbineVR. The tasks implemented are relatively less *interactive* than in FiskeVR. In WindturbineVR there is a more strict procedure, and a lot of the tasks are to switch buttons or inspect components, rather than actively picking up fish, sorting them, and cutting them. It seems that it is more important for users that the tasks engage them, rather than just understanding the tasks, which is quite natural. After all, this is what game literature says about the case; games are engaging because the tasks are engaging and challenging. The focus on usability, guiding, and presentation of the tasks are less important than the tasks being engaging. However, guiding and tutorials should not be avoided. For example, when adding a detailed interactive tutorial on climbing the ladder inside the tower, most users grasped the climbing much faster and quickly understood the concept of climbing in VR. The type of interactive tutorial that is employed at the ladder climbing in WindturbineVR, seems like the way the users grasp a concept faster, especially when animations are included.

It is important to note that the tasks for workplaces will vary, and some jobs, like wind turbine technician, does involve a lot of procedures, inspections, and pressing buttons. A continued balance between guiding, correct representation concerning the real job, and engaging tasks needs to be employed further in the project.

An important part of adding feedback and general information to the application is to not *overdo* it. Currently, there is no choice in the amount of information or help the user want in the application, and every guide and hint is shown to the user at all times. In the future, the information supplied and guides made can be even more aware of the user’s position and preferences, in order to only provide help when needed and wanted.

9.6 Using VRTK for development

This discussion relates to **RQ4:** "What advantages and disadvantages does Virtual Reality Toolkit bring to the development of a virtual internship?"

To experiment, and possibly achieve a faster development cycle, Virtual Reality Toolkit (VRKT) was used as the foundational SDK for development. VRTK helped achieve rapid development with its utility scripts, community, examples, and pre-made solutions for must use cases in VR: locomotion, interaction, user interface and more special utility like controller hints and highlighters. When developing features like climbing, snapping objects to certain places, teleporting, and interaction with objects, VRTK has been extremely valuable. With VRTK, the development process was rather quick, while still producing roughly similar results as FiskeVR in the evaluation tests (see Section 9.2). WindturbineVR was developed over the course of 3,5 months, while the previous virtual internship had been in development with several developers for ca. 10 months at the time of the evaluation test (30th of October, 2018). Of course, the rapidity of the development of WindturbineVR was helped by borrowing already established ideas and concepts from FiskeVR, not only due to VRTK. It should also be noted that this comparison of time is not very accurate or specific since it does not say anything about spent working hours on the applications and not anything about the amount of content created.

In addition to a massive amount of utility code VRTK includes proper documentation and an active and helpful online community on Slack (`https://vrtk-slack-invite.herokuapp.com/`). The examples included in the VRTK software is also beneficial, where almost every thinkable use case of the bundled code is applied and showcased. When a developer unfamiliar with developing for VR, these examples can be useful to build upon in the beginning. It should also be mentioned that VRTK is open source, and if there are any problems identified they can be fixed!

On the other hand, when adding a new dependency to a project, an additional external factor is suddenly vital to the project. An update of the underlying SDKs that VRTK relies on can break VRTK, or require an update to VRTK. Maybe an update can be slow to arrive or not directly compatible with the implementations in the developed application. In short, more complexity is added to the application, so if the utility of this added complexity is not high enough, it should be refrained from being added. Another downside was that virtually no code from FiskeVR could be reused in WindturbineVR. The usage of a different SDK between the applications also naturally results in some inconsistencies in implementation details and presentation of details to the user. For example, the default teleport pointer is not rendered the same in SteamVR and VRTK. These are inconsistencies that should not occur across the virtual internship project as a whole. Another point to make is that it could be that the perceived utility of all the examples and bundled code in VRTK is biased since the author had no prior experience in VR development and no extensive experience in game development. If seasoned VR developers were tasked with developing such an application, it could be that the experienced utility of VRTK is lower than what is experienced in this project.

There are many positives in using VRTK for development, from the fact that it is open source, provides loads of finished utility scripts, to the documentation and helpful community. However, there are also several downsides in adding such an external dependency to the application. At this point, it cannot be established if VRTK is the way to go further for the project as a whole, but what can be established is that the virtual internship project should eventually use the same SDK across all applications. A common SDK will reduce inconsistencies and double work on future applications. Since no common codebase yet exists between the different applications in the project, there is still a chance to choose a foundation for the project as a whole and build standard systems upon that, which can be used by all of the applications. A common foundation is one of the most vital points to bring the project further. This will reduce inconsistencies massively, reduce development time, reduce maintenance work, and in general increase the quality as functionality can be made one time thoroughly, not several times with different implementations. What should not be a continued practice is that each new developed workplace starts building everything from scratch. What the base SDK for the foundation should be is up for grabs, but VRTK is one to consider. The development of a common code base for the whole project will be a crucial part in developing the methodology of creating virtual internships.

Chapter 10

Conclusion

This thesis has both looked at relevant literature and an existing virtual internship to gain knowledge on the subject and identify areas of improvement. A new virtual internship has been developed, where the user can experience the work of a wind turbine technician. A virtual internship is a complex concept, including several areas and disciplines. It is a serious game, aiming for an affective learning outcome where the user will gain knowledge on whether a specific career could fit for them or not. It utilizes unfamiliar (for most) technology, and the usability in the application is an important focus.

Additionally, gaming elements like challenge, curiosity, feedback, and fantasy is essential. These elements must be regarded and implemented with care, always keeping in mind NAV's usage of the applications. Implementing engaging tasks, high-quality graphics, and appropriate sounds is essential to enable presence for the user, and this should be further improved upon in future versions. Several parts of the developed application were done differently than earlier versions: a new menu system, notifications on task completion, more guiding and in-game tutorials, and a dedicated teleport tutorial. The increased focus on tutorials and guiding did not increase the perceived usefulness, but it did reduce the amount of explanation that the users needed by the supervisor of the test and the perceived *ease of use*. These can be useful features when these applications arrive at NAV offices.

Further, a more detailed specification of the process from data and information to the finished workplace has been described and utilized. The process enabled rapid development of the workplace simulation, with five concrete steps from reality to virtuality, and has the potential to be the foundation of how the development should happen in future applications. With a more specific method, the developers and designers can identify more clearly what needs to be done in order to bring a particular part, or component of the workplace into the application, with a more controlled process. The software development methodology for an application as a whole should still follow a traditional agile development methodology, with iterations and deliveries along the way in order to identify problems and validate solutions.

Additionally, a different than previously used software development kit (VRTK) was used as the foundation in the development, which proved to be extremely useful. Whether this should be used or not for other, new or existing, applications cannot at this point be established. However, it is clear that it has helped tremendously to rapidly develop the application with all the included features, which reduced the amount of code and features that needed to be manually written by the author. The main takeaway though is that a common Virtual Internship SDK should be created; what this builds upon is not so important, but it is essential to create, in order to further speed up development and standardize how the user interacts with all these applications. From the results, and experience in using VRTK, it is emphasized as a valid contender for the foundation of a Virtual Internship SDK.

In addition to a common SDK for the project, future development in cooperation is the most exciting feature, both for the utility of the applications, but also from the users' point of view. It is a frequently requested feature, scoring relatively high in questionnaires, in addition to being highly relevant for certain workplaces. Finally, for future validation of the concept, the applications should be deployed at NAV offices so that employees can test these out as tools for career guidance. Data collection from the usage at the offices is vital to establish the value of the concept further.

This chapter will moreover present Research Questions, with brief answers in (Section 10.1), Recommendations in (Section 10.2), and finally Future Work in (Section 10.3).

10.1 Research Questions

This section presents the research questions and their answers as brief as possible. A longer discussion of these is presented in the discussion in Chapter 9.

RQ1: What is the current value of virtual internships? The current value of a virtual internship is its ability to engage, inform, and motivate young job seekers towards learning about a workplace and a job. It is not about training at workplace tasks, instead learn about the workplace, affecting the inner state of the user. The question ”Is this a job for me?” should be at least a little clearer after experiencing it in VR. Users feel it can be a valuable way to explore new jobs, and NAV employees feel it can be a valuable tool for them.

RQ2: What tools and processes are useful and necessary for creating a virtual internship and representing a workplace realistically? For each component or part of the virtual internship, there are, in reality, five essential steps, or processes that need to happen. First, there is *data* gathering about the component that is to be recreated. The data is information about the visual, physical, behavioral, and technical properties. For data gathering, research on the internet and contacting professionals are viable methods, and contact with the industry is vital. When enough data is gathered, the *digital* *representation* can be constructed, using a 3D model program. If a digital representation is already present and attainable without manual work, then this step is not needed. When creating the model, it is useful to create several versions with different levels of detail, to enable a dynamic *Level of Detail* in the game engine the model will be imported into, in order to secure adequate performance in the application. During the design of the model, it is crucial to keep in mind the functionality while creating it. After the model is created, *visual* aspects can be added. This includes UV-mapping and texturing. UV-mapping is important both when applying textures and normal maps, and when adding baked lighting in the game engine. When the model is ready *interactivity* can be applied, adding physics, colliders, properties, and necessary scripts so the player can interact with the component in the desired way. If the model is a static piece of the workplace, this is not needed. If the component is a part of or relates to a *task*, then logic to include it in that task is necessary.

RQ3: How do more focus on information, in-game guiding, and direct feedback impact the perceived usefulness and user-friendliness of a virtual internship? Generally, users rate the applications *ease of use*, higher than previous virtual internships. The increased focus on information and feedback makes the application more approachable, and less explanation is needed from a supervisor. The tutorial introducing the concept of teleporting has made the explanation of that concept much more manageable. Also, the immediate notification when completing tasks makes it visible to users that they can continue to the next task. These features do not make the applications perceived usefulness any higher, according to results and observations, but can be useful for general user-friendliness and to make it easier for NAV employees to use the applications as tools.

RQ4: What advantages and disadvantages does Virtual Reality Toolkit bring to the development of a virtual internship?

- **Advantages**: Virtual Reality Toolkit (VRKT) enables rapid prototyping and development due to the amount of bundled features, active community, and great examples. Also, it is platform agnostic and acts as an abstractor for other VR software development kits.

- **Disadvantages**: It adds another external dependency, and updates to underlying software can require an update to VRTK, which again can be incompatible with project-specific implementations. It increases complexity and makes it harder to reuse code previously made in earlier virtual internships as these are built on other foundations.

10.2 Recommendations

There is not substantially specific information in the literature on using virtual reality for career guidance and recruitment the way the virtual internship project does. This section will present the recommendations based on the result of this thesis concerning the recommendations that were presented in the Introduction (Chapter 1).

- **Game elements**: Further utilize and implement interactive tutorials, in-game guiding, and information that show up when needed. Balance the number of game elements and realism, and keep in mind the division between workplace-training applications and virtual internships.

- **More workplaces**: The practice of finding local companies in Trøndelag could prove a challenge in deploying these applications to the entire country. Can focus on more traditional, universal workplaces that is relevant in several areas of Norway.

- **Correct information**: The practice of contacting the industry to get information for the applications is valuable and should be continued.

- **Methodology**: A more detailed process like the five-step digitization process explained in this thesis should be used (Section 9.4). Development of a common code base is vital for future applications.
- **Training of NAV employees**: Employ the applications at NAV offices and let employees use them with the users they are responsible for.

10.3 Future Work

Future work and possibilities for the project are summarized here. This is not specific recommendations as given in Section 10.2, but more general recommendations and major future opportunities for the virtual internship project as a whole, as seen by the author. Future work is listed below:

- **Solid foundation**: Develop a common code base, solidly implemented in order to uniform the experience of the applications across the project, standardize implementations, and speed up development time.
- **Cooperation**: Cooperation with two or more users in an application is relevant for many lines of work where communication or cooperation with humans is essential. This feature can be used to "roleplay" situations as well, where one user is, for example, a customer and another is the cashier in a store.
- **Test in the field**: The applications should be deployed to NAV offices and tested in a more natural environment.
- **A common tutorial**: If all the applications use the same general concepts for interaction, then a common tutorial can be developed, that can be used for all applications. It is better, and easier to learn how to play through an interactive tutorial, rather than lengthy explanations from a supervisor or textual descriptions.

Bibliography

Accenture (2018). *ACCENTURE EXTENDED REALITY*. URL: https://www.accenture.com/no-en/service-extended-reality. accessed: 09.10.2018.

Ayala García, Andrés et al. (2016). "Virtual reality training system for maintenance and operation of high-voltage overhead power lines". In: *Virtual Reality* 20.1, pp. 27–40. ISSN: 1434-9957. DOI: 10.1007/s10055-015-0280-6. URL: https://doi.org/10.1007/s10055-015-0280-6.

Blender Foundation (2019a). *About*. URL: https://www.blender.org/about/. accessed: 25.03.2019.

— (2019b). *About UVs*. URL: https://docs.blender.org/manual/en/latest/editors/uv_image/uv/editing/unwrapping/introduction.html#about-uvs. accessed: 25.03.2019.

— (2019c). *Mapping Types*. URL: https://docs.blender.org/manual/en/latest/editors/uv_image/uv/editing/unwrapping/mapping_types.html. accessed: 25.03.2019.

Burdea, Grigore C. and Philippe Coiffet (2003). *Virtual Reality Technology*. Hoboken, New Jersey: John Wiley & Sons, Inc., pp. 1–6. ISBN: 0-471-36089-9.

Cohn, Mike (2014). *Agile needs to be both iterative and incremental*. URL: https://www.mountaingoatsoftware.com/blog/agile-needs-to-be-both-iterative-and-incremental. accessed: 02.06.2019.

Csikszentmihalyi, Mihaly (2014). “Toward a psychology of optimal experience”. In: *Flow and the foundations of positive psychology*. Springer, pp. 209–226. URL: https://www.ode.state.or.us/opportunities/grants/nclb/title_i/a_basicprograms/schoolimprovement/transformation7flow.pdf.

ESA (2015). *Essential Facts about the computer and video game industry*. online. URL: http://www.theesa.com/wp-content/uploads/2015/04/ESA-Essential-Facts-2015.pdf.

Fogden, Tom (2018). *Why is VR Failing to Take Off (Again)?* URL: https://tech.co/news/why-is-vr-failing-to-take-off-again-2018-08. accessed: 24.04.2019.

Fominykh, Mikhail and Ekaterina Prasolova-Førland (2019). “Job Interview Training in Virtual Reality: Evaluation in Laboratory Settings”. in 5th Annual International Conference of the Immersive Learning Research Network, 24-27 June, 2019, London, UK (in press).

Friess, Marianne Åsheim (2019). *Lærer av kollegaer sine gode erfaringar*. URL: https://memu.no/artikler/laerer-av-kollegaer-sine-gode-erfaringar/. accessed: 01.05.2019.

Gee, James Paul (2003). “What video games have to teach us about learning and literacy”. In: *Computers in Entertainment (CIE)* 1.1, pp. 20–20.

Good Energy (2018). *How do wind turbines work?* URL: https://www.goodenergy.co.uk/how-do-wind-turbines-work/. accessed: 23.05.2019.

Google (2017). *Designing Screen Interfaces for VR (Google I/O 2017)*. From a Goolge I/O 2017 Talk on Virtual Reality User interfaces. URL: https://www.youtube.com/watch?v=ES9jArHRFHQ.

Gottfredson, Linda S (2005). “Using Gottfredson’s theory of circumscription and compromise in career guidance and counseling”. In: *Career development and counseling: Putting theory and research to work*, pp. 71–100.

Haines, Eric (2012). *HeightmapFromTexture*. URL: http://wiki.unity3d.com/index.php/HeightmapFromTexture. accessed: 25.01.2019.

Heaney, David (2019). *VRTK v4 Beta Makes The Best Unity VR Framework Even Better*. URL: https://uploadvr.com/vrtk-v4-beta-release/. accessed: 21.04.2019.

Hybertsen, John (1998). "Statoil testing virtual reality on real seismic data". In: *Oil & Gas Journal* 58. URL: https://www.offshore-mag.com/articles/print/volume-58/issue-5/departments/technology-focus/statoil-testing-virtual-reality-on-real-seismic-data.html.

International Society for Presence Research (2000). *The Concept of Presence: Explication Statement*. URL: http://ispr.info/about-presence-2/about-presence/. accessed: 07.11.2018.

Isbister, Katherine (2016). *How games move us: Emotion by design*. Mit Press.

Jehan, Daniel Le (2018). *The future of extended reality in business*. URL: https://www.telegraph.co.uk/business/essential-insights/future-of-virtual-reality-in-business/. accessed: 09.10.2018.

Jerald, Jason (2016). *The VR Book: Human-Centered Design for Virtual Reality*. New York, NY, USA: Association for Computing Machinery and Morgan Claypool. ISBN: 978-1-97000-112-9.

John, Riya (2018). *Why augmented reality is important?* URL: https://medium.com/@riyajohn9495/why-augmented-reality-is-important-5f558fab2a0f. accessed: 05.04.2019.

Kirsh, David (1997). "Interactivity and multimedia interfaces". In: *Instructional Science* 25.2, pp. 79–96. ISSN: 1573-1952. DOI: 10.1023/A:1002915430871. URL: https://doi.org/10.1023/A:1002915430871.

Kizil, MS and J Joy (2001). "What can virtual reality do for safety". In: *University of Queensland, St. Lucia QLD*.

Klein, Heinz K. and Michael D. Myers (1999). "A Set of Principles for Conducting and Evaluating Interpretive Field Studies in Information Systems". In: *MIS Quarterly* 23.1, pp. 67–93. ISSN: 02767783. URL: http://www.jstor.org/stable/249410.

Lowenstein, George (1994). "The Psychology of Curiosity: A Review and Reinterpretation". In: *Exotic Preferences: Behavioral Economics and Human Motivation*. New York: Oxford University Press, pp. 121–181.

Malone, Thomas W. (1980). "What Makes Things Fun to Learn? Heuristics for Designing Instructional Computer Games". In: *Proceedings of the 3rd ACM SIGSMALL Symposium and the First SIGPC Symposium on Small Systems*. SIGSMALL '80. Palo Alto, California, USA: ACM, pp. 162–169. ISBN: 0-89791-024-9. DOI: 10.1145/800088.802839. URL: http://doi.acm.org/10.1145/800088.802839.

McLaren, Samantha (2017). *8 Innovative Ways Companies Are Using Virtual Reality to Recruit*. URL: https://business.linkedin.com/talent-solutions/blog/future-of-recruiting/2017/8-innovative-ways-companies-are-using-virtual-reality-to-recruit. accessed: 03.05.2019.

McLeod, Saul (2008). *Likert Scale*. URL: https://www.simplypsychology.org/likert-scale.html. accessed: 08.11.2018.

Milgram, Paul et al. (1995). "Augmented reality: a class of displays on the reality-virtuality continuum". In: vol. 2351, pp. 282–292. DOI: 10.1117/12.197321. URL: https://doi.org/10.1117/12.197321.

Mitgutsch, Konstantin and Narda Alvarado (2012). "Purposeful by design?: a serious game design assessment framework". In: *Proceedings of the International Conference on the foundations of digital games*. ACM, pp. 121–128.

Nguyen, V. T. and T. Dang (2017). "Setting up Virtual Reality and Augmented Reality Learning Environment in Unity". In: *2017 IEEE International Symposium on Mixed and Augmented Reality (ISMAR-Adjunct)*, pp. 315–320. DOI: 10.1109/ISMAR-Adjunct.2017.97.

Oates, Briony J (2006). *Researching Information Systems and Computing*. Sage Publications Ltd. ISBN: 1412902231.

Øygardslia, Kristine et al. (2018). *Virtuell Praksisplass: Evaluering av Fase 1*.

Prasolova-Førland, Ekaterina, Mikhail Fominykh, and Oskar Ekelund (2019). "Empowering Young Job Seekers with Virtual Reality". Submitted to the IEEE Conference on Virtual Reality and 3D User Interfaces 2019.

Procedural Worlds (n.d.). *Gaia*. URL: http://www.procedural-worlds.com/gaia/. accessed: 25.02.2019.

Qualcomm (2017). *The Mobile Future of eXtended Reality (XR)*. URL: https://www.qualcomm.com/media/documents/files/the-mobile-future-of-extended-reality-xr.pdf. accessed: 09.10.2018.

Reeves, Byron (1991). "Being there:" Television as symbolic versus natural experience." Unpublished manuscript, Stanford University, Institute for Communication Research, Stanford , CA.

Richey, Steve (2018). "Utilizing Presence in Augmented-Reality Applications to Improve Learning Outcomes". In: *Performance Improvement* 57.4, pp. 10–18. DOI: 10.1002/pfi.21773. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/pfi.21773. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/pfi.21773.

Seo, Jinseok and Gerard Jounghyun Kim (2002). "Design for Presence: A Structured Approach to Virtual Reality System Design". In: *Presence: Teleoperators and Virtual Environments* 11.4, pp. 378–403. DOI: 10.1162/105474602760204291. eprint: https://doi.org/10.1162/105474602760204291. URL: https://doi.org/10.1162/105474602760204291.

Sharples, Sarah et al. (2008). "Virtual reality induced symptoms and effects (VRISE): Comparison of head mounted display (HMD), desktop and projection display systems". In: *Displays* 29.2. Health and Safety Aspects of Visual Displays, pp. 58–69. ISSN: 0141-9382. DOI: https://doi.org/10.1016/j.displa.2007.09.005. URL: http://www.sciencedirect.com/science/article/pii/S014193820700100X.

SSB (2009). *Befolkning: Et aldrende samfunn*. URL: https://forskning.no/bakgrunn-demografi-partner/bakgrunn-befolkning-et-aldrende-samfunn/970893. accessed: 26.10.2018.

SSB (2018). *Vindkraft Øker Mest.* URL: https://www.ssb.no/energi-og-industri/artikler-og-publikasjoner/vindkraft-oker-mest. accessed: 03.12.2018.

Statkraft (2018). *Fosen Vind.* URL: https://www.statkraft.com/globalassets/1-statkraft-public/1-about-statkraft/projects/norway/fosen/factsheet-windfarms-fosen-vind-uk-2018.pdf. accessed: 03.12.2018.

Steuer, Jonathan (1993). "Defining Virtual Reality: Dimensions Determining Telepresence". In: *Journal of Communication* 42.4, pp. 73–93. DOI: 10.1111/j.1460-2466.1992.tb00812.x. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1460-2466.1992.tb00812.x. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1460-2466.1992.tb00812.x.

Summerson, Cameron (2018). *What Are the ARCore and ARKit Augmented Reality Frameworks?* URL: https://www.howtogeek.com/348445/what-are-the-arcore-and-arkit-augmented-reality-frameworks/. accessed: 04.10.2018.

Sweetser, Penelope and Peta Wyeth (2005). "GameFlow: a model for evaluating player enjoyment in games". In: *Computers in Entertainment (CIE)* 3.3, pp. 3–3.

Thomson, Stuart (2019). *YouGov: VR adoption grows to 11% of US adults.* URL: https://www.digitaltveurope.com/2019/03/22/yougov-vr-adoption-grows-to-11-of-us-adults/. accessed: 24.04.2019.

TrønderEnergi (n.d.). *Vindkrafttekniker.* URL: https://tronderenergi.no/om-tronderenergi/karriere/vindkrafttekniker. accessed: 25.05.2019.

Unity (2017). *Creating Beliavable Visuals: Where to Start?* URL: https://unity3d.com/learn/tutorials/projects/creating-believable-visuals/where-start. accessed: 25.02.2019.

— (2018a). *Baked lighting.* URL: https://docs.unity3d.com/2018.3/Documentation/Manual/LightMode-Baked.html. accessed: 23.01.2019.

— (2018b). *Colliders.* URL: https://docs.unity3d.com/2018.3/Documentation/ScriptReference/Collider.html. accessed: 25.04.2019.

— (2018c). *Color Grading*. URL: https://github.com/Unity-Technologies/PostProcessing/wiki/Color-Grading. accessed: 10.05.2019.

— (2018d). *Heightmap*. URL: https://docs.unity3d.com/Manual/StandardShaderMaterialPara html. accessed: 25.02.2019.

— (2018e). *LOD: Level of Detail*. URL: https://docs.unity3d.com/2018.3/Documentation/Manual/LevelOfDetail.html. accessed: 27.04.2019.

— (2018f). *Post Processing*. URL: https://github.com/Unity-Technologies/PostProcessing/wiki. accessed: 10.05.2019.

— (2018g). *Prefabs*. URL: https://docs.unity3d.com/2018.3/Documentation/Manual/Prefabs.html. accessed: 04.06.2019.

— (2018h). *Realtime lighting*. URL: https://docs.unity3d.com/2018.3/Documentation/Manual/LightMode-Realtime.html. accessed: 23.01.2019.

— (2018i). *Terrain*. URL: https://docs.unity3d.com/Manual/script-Terrain.html. accessed: 28.02.2019.

— (2018j). *Using particle systems in Unity*. URL: https://docs.unity3d.com/2018.3/Documentation/Manual/PartSysUsage.html. accessed: 10.05.2019.

— (2018k). *VideoPlayer*. URL: https://docs.unity3d.com/2018.3/Documentation/ScriptReference/Video.VideoPlayer.html. accessed: 01.05.2019.

Villani, D. et al. (2012). "May I experience more presence in doing the same thing in virtual reality than in reality? An answer from a simulated job interview". In: *Interacting with Computers* 24.4. Special Issue on Presence and Interaction, pp. 265–272. ISSN: 0953-5438. DOI: https://doi.org/10.1016/j.intcom.2012.04.008. URL: http://www.sciencedirect.com/science/article/pii/S0953543812000422.

VRTK (2018a). *VRTK_BezierPointerRenderer*. URL: https://vrtoolkit.readme.io/docs/vrtk_bezierpointerrenderer. accessed: 20.01.2019.

— (2018b). *VRTK_ClimbableGrabAttach*. URL: https://vrtoolkit.readme.io/docs/vrtk_climbablegrabattach. accessed: 20.05.2019.

VRTK (2018c). *VRTK_DashTeleporter*. URL: https://vrtoolkit.readme.io/docs/vrtk_dashteleport. accessed: 20.01.2019.

— (2018d). *VRTK_HeightAdjustTeleport*. URL: https://vrtoolkit.readme.io/docs/vrtk_heightadjustteleport. accessed: 25.04.2019.

— (2018e). *VRTK_InteractableObject*. URL: https://vrtoolkit.readme.io/docs/vrtk_interactableobject. accessed: 25.04.2019.

— (2018f). *VRTK_PlayerClimb*. URL: https://vrtoolkit.readme.io/docs/vrtk_playerclimb. accessed: 20.05.2019.

— (2018g). *VRTK_Pointer*. URL: https://vrtoolkit.readme.io/docs/vrtk_pointer. accessed: 20.01.2019.

Windpower Engineering (2011). *What are the electrical & electronic components in a wind turbine?* URL: https://www.windpowerengineering.com/electrical/electrical-electronic-components/. accessed: 10.05.2019.

Wouters, Pieter, Erik D Van der Spek, and Herre Van Oostendorp (2009). “Current practices in serious game research: A review from a learning outcomes perspective”. In: *Games-based learning advancements for multi-sensory human computer interfaces: techniques and effective practices*. IGI Global, pp. 232–250.

WWEA (2018). *WIND POWER CAPACITY REACHES 539 GW, 52,6 GW ADDED IN 2017*. URL: https://wwindea.org/blog/2018/02/12/2017-statistics/. accessed: 03.12.2018.

Wyk, Etienne van and Ruth de Villiers (2009). “Virtual Reality Training Applications for the Mining Industry”. In: *Proceedings of the 6th International Conference on Computer Graphics, Virtual Reality, Visualisation and Interaction in Africa*. AFRIGRAPH ’09. Pretoria, South Africa: ACM, pp. 53–63. ISBN: 978-1-60558-428-7. DOI: 10.1145/1503454.1503465. URL: http://doi.acm.org/10.1145/1503454.1503465.

Zyda, M. (2005). “From visual simulation to virtual reality to games”. In: *Computer* 38.9, pp. 25–32. ISSN: 0018-9162. DOI: 10.1109/MC.2005.297.

Appendices

.1 A1 - Unity Assets

This section is to bring credit to all assets used from the Unity Asset Store. Without the Unity community and the vast amount of tutorials and content available online this thesis would not be possible.

Name	**Description**	**Creator**
Ultra Skybox Fog	Fog effect for skybox	Galactic Studios
Living Birds	Automatically animated brids	Dinopunch
Yughues Free Concrete Barriers	Concrete barriers	Nobiax / Yughues
Serialized Dictionary	Serialiazable end inspectable dictionary	Rotary Heart
Low Poly Europan City Pack	Buildings and props for city	Karboosx
Bodyguards	Characters	Batewar
Microsplat	Terrain shading system	Jason Booth
Rock and Boulders 2	Realistic rocks	Manufactura K4
Classic Skybox	Skybox Materials	MGSVEVO
EasyRoads 3D Free v3	Road building system	Andasoft
Gaia	Terrain and scene generation	Procedural Worlds

Table 1: Assets from the Unity Asset store used in the application.

.2 A2 - NSD Form

Taking part in the research project

"Immersive Technologies for Learning and Training"

This is an inquiry about participation in a research project where the main purpose is to to explore the potentials and limitations of Immersive Technologies (virtual/mixed/augmented reality, VR/MR/AR) for learning and training in different areas, as a part of master student projects at Innovative Technologies for Learning (IMTEL) VR lab. To conduct this research, we will need to investigate the development and use of immersive technologies for learning and training in various contexts, including learning of language and mathematics, visualization of climate change, immersive exploration of historical manuscripts, workplace training and visualization of medical procedures. In this letter we will give you information about the purpose of the project and what your participation will involve.

Purpose of the project
To conduct this research, we will need to analyze the use immersive technologies for learning and training in various contexts, including learning of language and mathematics, visualization of climate change, immersive exploration of historical manuscripts, workplace training and visualization of medical procedures. The goal is to develop innovative learning methods and tools using immersive technologies.

Who is responsible for the research project?
NTNU, Department of Education and Lifelong learning is the institution responsible for the project.

Why are you being asked to participate?
You are asked to participate because you are a potential user of educational applications developed as a part of this project and have visited our lab/expressed interest in immersive technologies. Your feedback is important for develop innovative learning methods and tools.

What does participation involve for you?
You will be ask to test immersive applications for learning and training purposes and then give feedbacks in the form of questionnaires and interviews/group interviews.

Participation is voluntary
Participation in the project is voluntary. If you chose to participate, you can withdraw your consent at any time without giving a reason. All information about you will then be made anonymous. There will be no negative consequences for you if you chose not to participate or later decide to withdraw.

Your personal privacy – how we will store and use your personal data
We will only use your personal data for the purpose(s) specified in this information letter. We will process your personal data confidentially and in accordance with data protection legislation (the General Data Protection Regulation and Personal Data Act). Any data that can be traced to individual participants will be kept confidential and anonymized before being used for research purposes. Parts of the sound recordings will be transcribed (written down) and stored electronically. All source data will be handled and stored in accordance with the existing regulations by NTNU as the responsible institution and only persons associated with the project (IMTEL VR lab research personnel and master students) will have access to them.

What will happen to your personal data at the end of the research project?
The project is scheduled to end 31.12.2019. All data will be anonymized at the end of the project, e.g. audio and video will be deleted when transcripts and analysis of data are completed, except for selected video and photo material to be used for research purpose. These and anonymized recordings from the inside of the virtual environments may be used for demonstrations in research context in such a way that no information will be linked to individuals. Scientific reports and presentations from this study might contain recordings from the VR/MR/AR sessions, questionnaire results, anonymized photos/videos from the sessions and anonymized citations from the interviews.

Your rights
So long as you can be identified in the collected data, you have the right to:
- access the personal data that is being processed about you
- request that your personal data is deleted
- request that incorrect personal data about you is corrected/rectified
- receive a copy of your personal data (data portability), and
- send a complaint to the Data Protection Officer or The Norwegian Data Protection Authority regarding the processing of your personal data

What gives us the right to process your personal data?
We will process your personal data based on your consent.

Based on an agreement with NTNU, NSD – The Norwegian Centre for Research Data AS has assessed that the processing of personal data in this project is in accordance with data protection legislation.

Where can I find out more?
If you have questions about the project, or want to exercise your rights, contact:
- Ekaterina Prasolova-Førland (Department of Education and Lifelong Learning, NTNU)
- phone: +47 99 44 08 61, email: ekaterip@ntnu.no
- NSD – The Norwegian Centre for Research Data AS, by email: (personverntjenester@nsd.no) or by telephone: +47 55 58 21 17.

Consent form

I have received and understood information about the project **Immersive Technologies for Learning and Training** and have been given the opportunity to ask questions. I hereby declare my consent that my data in relation to Immersive Technologies for Learning and Training may be stored, documented and used for research and educational purposes as described above. I give consent for my personal data to be processed until the end date of the project, approx. 31.12.2019

(Signed by participant, date)

Jørgen Henrichsen

Engaging Young Job Seekers with a Wind Turbine Technician Internship in Virtual Reality

NTNU
Norwegian University of
Science and Technology