Master's thesis

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

Aleksander Johansen & Eivind Alfsvaag Johansen

Enhancing System Usability for Immersive Job Taste with CarpenterVR

Master's thesis in Computer Science
Supervisor: Monica Divitini
Co-supervisor: Ekaterina Prasolova-Førland

June 2022

NTNU
Norwegian University of
Science and Technology

Aleksander Johansen & Eivind Alfsvaag Johansen

Enhancing System Usability for Immersive Job Taste with CarpenterVR

Master’s thesis in Computer Science
Supervisor: Monica Divitini
Co-supervisor: Ekaterina Prasolova-Førland
June 2022

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

NTNU
Norwegian University of
Science and Technology

Enhancing System Usability for Immersive Job Taste with CarpenterVR

Aleksander Johansen
Eivind Alfsvaag Johansen

June 20, 2022

Acknowledgement

We want to thank our supervisors, Monica Divitini and Ekaterina Prasolova-Førland, for guiding and helping us by providing invaluable information, feedback on the app and the thesis, and users for testing our application. We also want to thank the NAV's career guidance employees, especially Heidi Fossen and Lars Bjønnes, for testing and providing end-users to test our application and all who tested our application. Mikhail Fominykh and everyone working at IMTEL lab for guiding us and providing invaluable information and support. Finally, we thank our friends and family for their support and encouragement.

Abstract

Unemployment rates have risen in the past few years due to the COVID-19 pandemic. Therefore, some solutions should be found to mitigate this trend. Virtual reality (VR) has seen a lot of prominence in recent decades, which provides new ways for entertainment and learning. The latter is especially useful in these times since it allows for a more engaging way for youth to learn and retrieve information. Information about different professions can often be vague or not descriptive enough. Relying entirely on text-based descriptions is usually not encouraging enough for young jobseekers when choosing what to pursue. Immersive Job Taste is a concept that seeks to empower and motivate young jobseekers with VR. Many VR experiences have been created for Immersive Job Taste, all with different occupations as a focus. This thesis has created a VR application based on the carpenter occupation and executed a case study to investigate usability in previous Immersive Job Taste applications. The focus of this thesis is how to improve the usability of the Immersive Job Taste applications. With SUS scores and observations, the carpenter application has been developed and tested between iterations with increasing SUS scores. SUS has been used to evaluate the VR application as it is a quick, reliable, and well-established way to test a system's usability.

Sammendrag

Arbeidsledigheten økte under Covid-19 pandemien, og det oppsto behov for å finne tiltak for å endre denne utviklingen. Virtuell Virkelighet (VR) har blitt mer tilgjengelig og anvendelig de siste ti-årene. Teknologien har fått nye bruksområder, også som virkemiddel for læring i tillegg til underholdning. Ikke minst er bruksområdene knyttet til læring vesentlig, fordi det åpner for nye og mer engasjerende måter å motta informasjon på. Å måtte basere seg utelukkende på skrevne beskrivelser vil ofte ikke være interessant nok for unge jobbsøkere når de skal velge hvilken vei de vil gå. «Immersive Job Taste» er et konsept som søker å styrke og motivere unge jobbsøkere i deres valg, ved bruk av VR. Mange VR-opplevelser er blitt laget for «Immersive Job Taste», med utgangspunkt i ulike yrker. Denne oppgaven har laget en VR-applikasjon basert på tømreryrket. I den forbindelse er det gjennomført en kasusstudie der målet var å forbedre brukervennligheten til «Immersive Job Taste» applikasjoner. Med SUS-skår og observasjoner, har tømrerapplikasjonen blitt utviklet og testet mellom iterasjoner med økende SUS-skår.

Contents

Figures

Tables

Acronyms

AR Augmented Reality. xv

EHS Environment, Health and Safety. xv

HMD Head Mounted Display. xv

IDI Department of Computer Science. xv

IMTEL Innovative Immersive Technologies for Learning. xv

NAV Norwegian Welfare Administration. xv

NES Nintendo Entertainment System. xv

NTNU Norwegian University of Science and Technology. xv

VR Virtual Reality. xv

XR Extended Reality. xv

Glossary

CSS CSS stands for Cascading Style Sheets. It is a coding language used to define of files written in HTML or XML. xv

Cybersickness Cybersickness is a form of motion sickness that occur in the absences of physical motion in VR. xv

Github GitHub is a code hosting platform for version control and collaboration. xv

Gitlab Gitlab is a code hosting platform for version control and collaboration.. xv

Likert scale A numeric scale containing values from 1-5 or 1-7 which corresponds to how much users agree with a particular statement. xv

PNG PNG is a file format used to save images. PNG is using lossless compression. xv

Scripts Scripting languages are computer code that is translated to computer instructions when run. xv

Unity Unity is a cross-platform game engine developed by Unity Technologies. xv

URL Universal Resource Locator is a complete web page address used to find a website. xv

Chapter 1

Introduction [1]

This master thesis created a VR experience. A video can be viewed showcasing it here: `https://youtu.be/jEqvKzEzvgk`

1.1 Context

This master thesis is part of the Virtual Internship project developed at Innovative Immersive Technologies for Learning VR lab (IMTEL) at the Norwegian University of Science and Technology (NTNU). The project is a collaboration between IMTEL, the Department of Computer Science (IDI), and the Norwegian Labor and Welfare Administration (NAV). The Virtual Internship is a collaboration between IMTEL and NAV. The project aims to help young jobseekers by using immersive and interactive technologies.

Immersive Job Taste is a concept formed from the Virtual Internship project. This concept targets young jobseekers and high schoolers searching for a job but doesn't know what career path they want to pursue. It is a presentation of occupations with elements of workplace training[1]. The purpose of Immersive Job Taste is for young jobseekers to try out different professions by performing everyday tasks found in various occupations to see if it is something that they would like to pursue further as a career choice. These immersive apps aim not necessarily to give jobseekers workplace training but rather an insight into the profession by affecting their internal state [2]. Developed Job Taste apps include fishery worker, wind turbine electrician, road construction worker, car mechanic, tinsmith, warehouse worker, crane operator, pharmacy technician, and dentist assistant. The Virtual Internship project also has a VR job interview app[3].

One of the goals of this thesis is to create a new Virtual Internship for IMTEL's catalog of Immersive Job Taste apps. The new app and its details were discussed

[1] This thesis includes content from the specialization project TDT4501 for Aleksander Johansen and IT3915 for Eivind Alfsvaag Johansen, both submitted in autumn 2021.

with employees at NAV and our supervisors at IMTEL to determine what profession it should be about. It was ultimately decided that carpentry would be the focus of the new app. The idea of having a carpenter Immersive Job Taste app was first the idea of an Experts in Teams(EiT) group project[4]. This project made a VR app allowing users to install wall plates.

An additional topic discussed was the usability and user interfaces of these apps. One of the common issues with pre-existing Job Taste apps is that they are not always easy to use or understand. Previous projects, specifically the car mechanic app and the crane app, detail in their respective master theses that usability is an aspect that can be improved upon [5][6]. These projects tested their apps regarding system usability, showing that their apps hover between average and below average. Due to these reasons, it was decided that usability would be an essential aspect of mind when designing and creating the new carpenter app.

As our goals for this thesis are twofold, there are several contributions for both IMTEL and NAV. Firstly is the research into usability for existing Immersive Job Taste apps, which will highlight essential aspects to make these apps as user-friendly as possible. Secondly is the new carpenter app and the different components it consists of. For example, the new system for displaying tasks or the language system are parts of code that can either be reused or taken as pieces of inspiration when improving existing apps or creating new ones. Another significant contribution is the results produced during user testing, which provides insightful information regarding users' thoughts about the system being built.

This master spanned from January 2022 to June 2022. In this time frame, research was done on earlier Job Taste apps, and a new app was created that NAV users, NAV employees, and visitors of the IMTEL lab tested.

1.2 Motivation

Due to the COVID-19 pandemic, the unemployment rate has steadily increased, especially in tourism and transportation, where had the unemployment rate quadrupled in Norway [7]. This has further raised the need to develop innovative workplace training and career guidance methods. To do this, we will explore how virtual reality (VR) can be used to guide young job seekers who are not sure want kind of profession they want to pursue. With VR, the participants can try different work tasks in various jobs to find out if the profession could be something they think is worth pursuing. There are multiple VR applications to try professions in the Immersive Job Taste catalog [3]. Some of these VR applications have been tested with SUS scores. The ones that have been tested have a score of around 63 – 71 SUS score. The average SUS score is 68[8]. This makes these have about average SUS scores. Both authors have tested most of the applications and had problems with understanding what they are supposed to do and some issues with the UI. Both wanted to understand what could be done to help make these apps more usable.

1.3 Research Questions

RQ1: How to improve System Usability with UI in Immersive Job Taste apps?

RQ2: What value does the carpenter app provide as an Immersive Job Taste?

RQ3: What guidelines should be considered when designing Immersive Job Taste apps to enhance the user experience?

Chapter 2

Theory

2.1 Virtual Reality

Virtual reality is defined as: “Inducing targeted behavior in an organism by using artificial sensory stimulation, while the organism has little or no awareness of the interference.”[9]. This definition has four main components: *targeted behavior*, *organism*, *artificial sensory stimulation*, and *awareness*. The targeted behavior covers the experiences the user receives from the creator. Organism refers to the living creature that is exposed to sensory stimulation. Artificial sensory stimulation is when one or more of the organism’s senses becomes co-opted, and their regular inputs are replaced or enhanced through artificial stimulation. The awareness is what “fools” the organism into feeling present in a virtual world. In other words, the organism is unaware of the interference caused by artificial stimulation. This unawareness leads to the feeling of being *present* in the altered/alternative world [9].

However, this definition of virtual reality is a bit broad and perhaps vague as it can be interpreted differently. One could argue that things like listening to music or watching a movie would fall under VR based on this definition since it provides artificial sensory stimulation. For our purposes, something more specific would be appropriate as we’re dealing with the use of HMD to immerse users into an alternate reality. According to Oxford Language, virtual reality is defined as: “the computer-generated simulation of a three-dimensional image or environment that can be interacted with in a seemingly real or physical way by a person using special electronic equipment, such as a helmet with a screen inside or gloves fitted with sensors.”. This definition is more suited for the kind of VR described in this thesis since we focus on using computers and HMDs to create a virtual world the user can experience and interact with.

2.2 Learning in VR

As VR has become more advanced, available, and affordable over the last decades, it has led to VR being used in several aspects of educational settings. VR is especially used for teaching concepts such as science and mathematics [10]. This is because these subjects are very theoretical and abstract, making them difficult to grasp for some people. However, using VR, these abstract, intangible concepts become more concrete, manipulable, and thus easier to understand. VR can also be used to explore subjects such as history, social studies, and foreign languages since users can immerse and therefore explore historical places they normally would not be able to [10]. VR has in the past been used as a tool to teach middle schoolers about gorillas and their lifestyles. Such as behaviors and social interactions [11]. This was very useful since this was something one could not learn simply by just visiting the zoo [11].

One of the reasons why VR is useful for learning is that it provides a 3D environment that is interactive and can give audio, visual, and haptic feedback [12]. VR is, therefore, very beneficial to use in subjects where visualization of the material is important. 3D visuals are not exclusive to VR, as creating videos or animations to accomplish the same visual feedback is possible. However, there is one crucial difference between a video or animation shown on a traditional display and an immersive VR application – interaction. The former is a passive way to teach, while the latter is interactive. Having interactive learning methods is something that promotes active learning instead of passive learning [12]. Active learning has the benefit that it can increase the retention rate of what has been taught. For instance, it has been shown that six months after a traditional lecture and text-based class, 90% of the curriculum was forgotten. At the same time, active learning was able to retain up to 70% after two years [13].

The use of VR can, in addition, to learning about educational subjects, also be used to teach people different operational skills in the work industry. There was research in which VR was used to train dental students [14]. It was shown that by using a VR simulation, students got improved scores on their tests and improved their operative dentist skills. This research shows that the students learned procedures faster with a VR simulation than those trained in traditional laboratories.

2.3 Computer Games

Definition of video games "*Video game*: a mental contest, played with a computer according to certain rules for amusement, recreation, or winning a stake." [15]
The first computer game created was most likely the game OXO by Alexander Douglas in 1952. Still, most consider *Tennis for Two* by William Higginbotham to be the first interactive computer game[16]. These games needed vast amounts of equipment to play, as seen on *Figure 2.1*, unlike the computer games today

that fit in your pocket *Figure 2.2*. Since the games in the 1950s, there has been an enormous development in computer games, equipment, and industry. The introduction of *Odyssey*, a home console that could be connected to a TV set, was the start of home consoles[17]. Since then, the developments in hardware, interaction devices, and software tools have had a massive effect on the possibilities of what is possible to do with computer games. The gaming console Nintendo Entertainment System (NES) was released in 1985,[18] and the Sega Master System in 1986[19]. The NES became the more successful console selling about 50 million more consoles, even though the NES was a less advanced technological console[18][19]. It was the quality and uniqueness of the games released with the NES that showed that it is the games that matter, not the technology behind them.

Figure 2.1: Tennis for Two (public domain Originator Brookhaven National Laboratory)

Figure 2.2: Gameboy micro (public domain Originator Brookhaven National Laboratory)

2.4 Human-Centred Design

New development often requires time before it adapts good design principles. This is where human-centered design can be used to find good design principles. Human-centered design is a process of incorporating the end-users in the design and development process. The process focuses on the capabilities and needs of the people they are intended for. "This approach enhances effectiveness and efficiency, improves human well-being, user satisfaction, accessibility, and sustainability; and counteracts possible adverse effects of use on human health, safety and performance."[20]. Traditional Human-centered design is an iterative process of observation, ideation, prototyping, and testing[21]. This makes it easy to test possible solutions to problems quickly. The discoverability of a product is essential to understanding how to interact with a product. Discovering its functions and operations results from appropriate application of five fundamental psychological concepts: *affordances*, *signifiers*, *mapping*, *feedback*, *conceptual models*, and *constraints*[21].

2.4.1 Affordances

In everyday life, people interact with many different objects, often new objects. How do they know how to open a new door they have never seen before? The answer is *affordances*. Affordance is a term that refers to the relationship between an interacting agent, whether human, animal, robot, or machine, and an object[21]. An affordance determines the possibilities an object can be used based on the capabilities of an interacting agent and the properties of the object. Objects might have many possible affordances depending on the interacting agent. A fridge has the affordance of opening, but this affordance is based on the capabilities of the interacting agent. A toddler might not be able to open the refrigerator, but a healthy adult might. In this case, the affordance *open* is only available for the adult.

Objects can also have anti-affordances. Glass is relatively invisible and is one of the main reasons for using glass, but it has an anti-affordance of blocking passage. The anti-affordance is not easily discoverable; hence birds and people crash into them. Affordances and anti-affordances that are not discoverable need some way of signaling their existence. This property is called *signifier*.

2.4.2 Signifiers

Signifiers communicate a possible affordance to the interacting agent. The designer uses them to help the user understand the possible functions of an object. These signifiers can be used intentionally and unintentionally[21]. An arrow that points to a button is an intentional signifier, but the presence or absence of people waiting at a bus stop to determine whether we have missed the bus is an unintentional signifier.

The reason signifiers are required is because not all design is easy and intuitive to understand. In the picture under, 2.3 there is no way of knowing which way to pull or push as the handles are the same on both sides. There are multiple ways to signify a door to be pushed or pulled. One way is to have two signs that say push and pull on the respective doors, and another way is to have different handles for the push and pull doors. Signaling with the door handles how to use them.

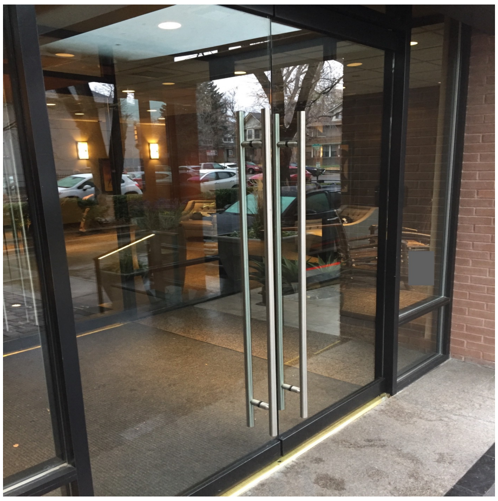


Figure 2.3: Door with the same bars on both sides.

2.4.3 Conceptual Models

Conceptional models are explanations of how things work[21]. They often are highly simplified and do not need to be accurate as long the model is useful. Take a web browser loading a website. A simplified conceptional model might be that when a URL is typed in the web browser, it goes to the webpage. This is not the entire web browser to webpage interaction model, just a simplification. There are several layers between entering a webpage domain into the web browser and loading the page. This model works for most everyday usage, but if the webpage server is down, the result can be confusing. People can have different conceptional models. An average internet surfer's conceptual model of the internet is quite different from an internet technician's conceptual model.

Conceptional models are created in the mind from different sources. They can be made by observing the device, explained from person to person, reading about the device, or from experience. Often the conceptional model has many errors and creates difficulties using the device[21]. The designer can help create conceptional models that allow the user to understand the device properly. The main tools are *affordances*, *signifiers*, *mapping*, *feedback*, and *constraints*[21]. Most conceptional models need not be complex. They help predict how things will behave and understand what to do when things do not work as intended. The most important is to understand the relationship between controls and outcomes.

2.4.4 Mappings

Quote: "Mapping is a technical term, borrowed from mathematics, meaning the relationship between the elements of two sets of things."[21]. When designing the layout of controls and displays, mapping is essential. This is because it helps people make a conceptual model of the object they are interacting with. These models do not need to be accurate in most cases. Conceptional models help the person remember how the actions affect the object. Adding spatial analogies to the mapping leads to immediate understanding[21]. For example, a joystick in a forklift moving the joystick forward moves the forklift forward, and moving the backward joystick moves the forklift backward. Here the only conceptual model needed is that the joystick moves the forklift. The person operating the forklift does not need to understand how it moves, just that it does.

2.4.5 Feedback

Feedback is another way of communicating with the interacting agent. Feedback communicates the result of actions taken. Feedback helps the interacting agent understand if he is doing the action correctly. A simple task like picking up an egg requires correct feedback. Using too much strength and the egg breaks. Using too little, and it falls. Most humans can feel how much force is needed with the help of feedback from their fingers. Any delay in feedback may cause unintentional consequences. In the best case, the interacting agent might do the action again with no consequence except the annoyance of the interactor. In other cases, the action might occur twice, cause an error, or not happen. Even a tenth of a second can be disconcerting[21].

The quality and quantity of feedback are essential. If all the feedback is the same, it can become confusing what is happening[21]. When a person gets too much feedback, it can be annoying and create a worse user experience. The way feedback is used needs to be intentionally designed. Different feedback is required for different signals. Some signals are more important than others. Thus the feedback needs to capture the user's attention. Other signals are less important and need a discreet way to give feedback.

2.4.6 Constraints

Constraints can be classified as **physical**, **cultural**, **semantic**, and **logical**[21].

Physical constraints are physical limitations that constrain possible operations. An example of this can be how most home doors have no physical constraints on people walking thru the door, but if a car tries to go thru the door, it physically can't. This constraint is best used when it is easy to see and interpret the physical constraint. This is because it reduces the possible actions before anything has been done. If it is not easy to see or interpretable, the physical constraint only prevents wrong actions after the action has been executed.

Cultural Constraints is a set of allowable actions based on the culture. These constraints are how we know how to behave and act in the cinema. These constraints are knowledge necessary to interpret situations.

Semantic Constraints depend on the meaning of the situation to control possible actions. This type of constraint depends on the knowledge of the situation and the world. In an elevator, the meaning of the situation might be to move from one floor to another. The meaningful action is to press the button corresponding to the floor intended. There are multiple actions, but only one meaningful action.

Logical Constraints are constraints that often show themselves after or during the execution of a task. A logical constraint can be having extra screws after disassembling and reassembling or the last piece of a puzzle. The first shows that the task has been done incorrectly, and the second shows the last possible placement for the last puzzle piece.

2.5 System Usability Scale

Brooke developed the System Usability Scale (SUS) as a “Quick and dirty usability scale” in 1996[22]. SUS was used to evaluate our application as it is a reliable, quick, and well-established way to test a system’s usability. The average SUS score is 68[8]. SUS is a 10-question survey given to the tester after testing but before any discussion of the system. Each question is answered using a Likert-5 score which gives a score between 1 and 5.

SUS score is calculated from a 0-40 scale, where for each odd-numbered question, subtract 1, and for each even-numbered question, take five and subtract the score. This is done because even-numbered questions are negative questions. When the SUS score is calculated to 0-40, it is multiplied by 2,5 to a 0-100 score. This is not a percentage but an easier way of viewing the score.

$$Odd = (Q1 + Q3 + Q5 + Q7 + Q9) - 5$$

$$Even = 25 - (Q2 + Q4 + Q6 + Q8 + Q10)$$

$$(Odd + Even) \times 2.5 = SUSscore$$

Scores can be divided into different grades:

SUS score	**Grade**
100-78.9	A
78.8 – 72.6	B
72.5 – 62.7	C
62.6 – 51.7	D

Table 2.1: Different grades based on SUS score[8]

Chapter 3

Related work

3.1 Related Career Guidance Apps

3.1.1 Immersive Job Taste: a Concept of Demonstrating Workplaces with Virtual Reality

The concept of “Immersive Job Taste” was presented in the paper *Immersive Job Taste: a Concept of Demonstrating Workplaces with Virtual Reality*[1]. The paper aimed to explore the potential of VR, AR, and other gaming technologies to motivate better and empower young job seekers. The paper created a methodology for the design of Immersive Job Taste applications. The methodology described how to create tasks and how to represent them. The representation of the tasks is supposed to be two parts. One part where the user watches a 360 video and another where the task is simulated. This is to let the user first observe and then try themselves. This paper created two applications: FisheryVR, which followed the methodology, and InterviewVR, which did not follow the method.

The FisheryVR is a VR application created for the HTC Vive and developed in Unity 3D. It has six typical tasks that users can try and receive feedback on each task. The tasks are split between the fish farming and fish processing facilities. The tasks used 360-degree videos to show and describe the tasks.

The InterviewVR application simulated an interview with training functions in VR. In the application, the user is placed in a 360-degree video with the interviewer. The user can answer the interviewer with a list of answers. There were 12 questions and 14 optional comments or questions as part of the interview. Both the questions from the interviewer and the response from the user are read aloud. When the user had finished the interview, they had an option to play back the interview.

Feedback from testing the applications was very positive. Both young jobseekers and the welfare professionals were mostly favorable towards the project and saw value in it. The concept of Immersive Job Taste help makes a more interactive way to learn about different professions. They can also be less time-consuming than visiting the companies in real life.

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

In the thesis *Engaging Young Job Seekers with an Internship as a Wind Turbine Technician in Virtual Reality*, VR was used as a tool for career guidance which is very similar to what we are doing [2]. For this project, a specific VR app was tested on different users. The app is called "FiskeVR" and simulates the fishing industry. In this app, the user is placed in a salmon farming facility. Here the user has to perform regular tasks that a worker has to do, such as feeding the fish, driving a boat to check cages of salmon, and processing the fish by sorting, packing, and filet cutting. This app was tested by several users, students, NAV employees, and young job seekers. The app was tested on several aspects, such as how enjoyable it was to use, how useful it was for receiving insight into the profession, and whether or not it should be used as a career guidance tool. The results from the participants were overall positive on these different aspects gaining around a score of 4 or higher on average on the Likert 5-point scale. The conclusion was that users generally felt that the application was helpful as it gave insight into the fishing industry and provided a moderate dose of realism. The NAV employees were even more enthusiastic about the application and requested that more workplaces other than the fish industry be implemented.

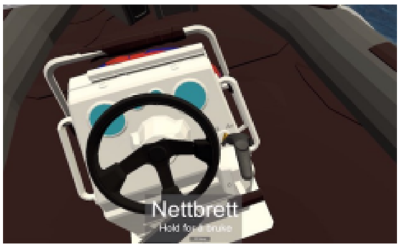


Figure 3.1: Boat steering wheel, ignition and throttle in FiskeVR

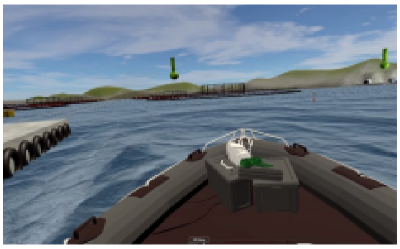


Figure 3.2: Boat and destination markers in FiskeVR

The second part of the thesis focused on creating another career guidance app in the form of a wind turbine technician simulation. In this application, the user performs regular tasks by a wind turbine technician. This project was developed in an iterative process where requirement specifications were defined, implemented, and then tested by various users (students, job seekers, and NAV employees). The feedback was pretty similar to "FiskeVR." However, people also tried the app in the wind turbine industry. The technicians felt that although the application simplifies their tasks, it gave good insight into the most common tasks that a wind turbine technician regularly does. On the other hand, they also commented that there was no option/possibility to cooperate and communicate with other technicians, which is an essential aspect of their job.

3.2 Career Labs VR - VR Career Experiences

Career Labs VR is an employment consultant located in Canada. One of their projects is called "VR Career Experiences» and is a project very similar to IMTEL's Virtual Internships. The project aims to accomplish much of the same: to let students and job seekers explore different career choices with the use of VR [23]. These VR apps are simulations of different professions and allow users to gain insight into various experiences on the job. Some of the occupations they include are stick welding, residential electrician, heavy equipment operator, process engineer, pipe fitter, and several more. There are other goals of this project besides providing the opportunity to explore various professions. One is training, where staff is trained and coached using VR to increase safety and improve results [23]. The final is to use it as a tool for recruitment. This allows different institutions such as schools, jobs, and career guidance centers to provide an alternative learning method.

3.3 Design of Virtual Reality Simulation-based Safety Training Workshop

In *Design of Virtual Reality Simulation-based Safety Training Workshop* they used VR to teach students about safety when handling power tools such as drills and circle saws[24]. The study aimed to create a learning medium that is more effective than a standard PowerPoint presentation about power tools while also letting them test the tools in a harmless environment to reduce incidents in the workshop. The VR application was made using Unity as a game engine and Blender for creating models (shown in figure 3.3). The participants could move around in this workshop and learn about the safety guidelines by reading instructions on the wall. They could then interact with the different tools to figure out how they work (how to turn on/off the tool and how to cut the material). In addition, a quiz game was implemented, which each participant had to answer during the training. This purpose was to familiarize the users with the different tools before testing them in real life. To test the effectiveness of this tool, the researchers performed an experiment where they had two groups of students (5 in each group) be trained using different methods. The first group would undergo regular training by attending a classroom lecture using PowerPoint slides, while the second group would use the VR workshop tool. After training using the different methods, the students were tested by making them answer 35 questions on what they learned. The results showed that the group using the VR workshop achieved a higher score on the test with an average of 71.4%. In contrast, the opposing group achieved an average result of 40% [24]. A survey questionnaire was also made to let the users give feedback on the application. Most responses were that VR is a good tool for workshop training. This was mainly because the virtual workshop was more interactive than the PowerPoint slides, which only provided a one-way communication line.

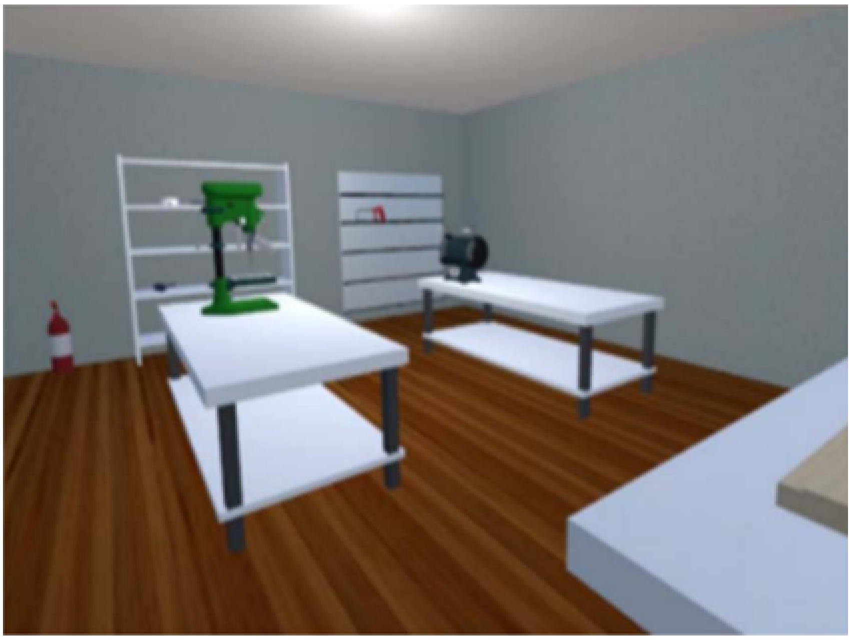


Figure 3.3: Virtual workshop with power tools

3.4 Effect of virtual reality technology on the teaching of urban railway vehicle engineering

The paper *Effect of virtual reality technology on the teaching of urban railway vehicle engineering* describes a virtual reality learning environment (VRLE) which was designed for training railway vehicle engineer students to assemble and dismantle a bogie [25]. This VRLE system includes three different modes: teaching, training, and testing. The teaching mode is made for explaining the theory behind the task of assembling/dismantling the bogie. This is done by having a teacher operating the VRLE system or using courseware included in the VRLE. The theory could also be obtained through educational videos in the VRLE library. The training mode lets users hone their skills in two categories – basic and operational skills. The basic skills cover the user's ability to recognize and apply the tools used for assembly/dismantling. The operational skills train the user to assemble and dismantle a bogie. Finally, the testing mode can evaluate the students by measuring the learning outcomes of their training. To test this VRLE system, they created two groups of 40 participants each (31 males and 9 females). One group trained using the traditional method, and the other used the VRLE system. After training, it showed that the group using VRLE achieved a vastly higher mean score than

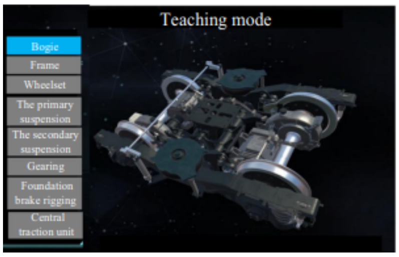
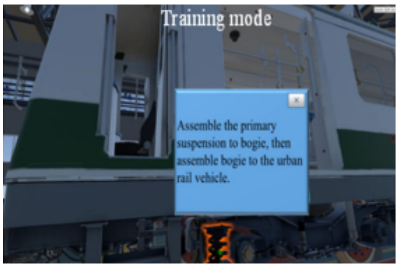


Figure 3.4: Teaching mode



Assemble the primary suspension to bogie, then assemble bogie to the urban rail vehicle.

Figure 3.5: Training mode

the group using traditional methods. The conclusion was that the VRLE system was more effective than traditional methods for training the students' operational skills.

3.5 VR UI Guidelines

In the paper *Evaluation of Graphical User Interfaces Guidelines for Virtual Reality Games* the researchers wanted to evaluate the main guidelines used by designers and programmers of Graphical User Interfaces (GUI) in VR games[26].

Five problems are identified in the paper when using GUI in VR:

1. Using 2D solutions
2. The depth of placing a 3D object
3. Distance between the player and the GUI
4. The way text is displayed.
5. The problem with opening a GUI and the GUI being occluded by the environment.

The guidelines tested are a combination of Oculus, Unity, and Leap Motion guidelines. There are eleven guidelines altogether if the overlapping guidelines are not counted. These are the recommended guidelines:

1. **Provide depth tip.** As there is no actual depth in a VR headset, it is hard to judge the depth of nearby objects. In the real world, our eyes dynamically assess the depth of nearby objects. Some of the tricks a developer can use include: Objects in the distance lose contrast, distant objects appear fuzzy, nearby objects appear sharp, and a shadow from a hand appears on objects[27].
2. **Comfortable content distance.** This guideline refers to important objects the user will be looking at and recommends a distance of 0.5m to 1m from the user's view.
3. **Incorporate the GUI into user's environment or character.** This refers to making GUIs a part of the world instead of an overlay on the HMD.
4. **Use text in UI that are easily read.** VR has a problem with text in general.

It is often hard to read if the text is not the correct size. It can appear blurry. It is crucial that the text is the right size and not blurry for the user to be able to read the text.

5. **GUI should surround the user.** This is, so the distance from the GUI stays the same. If the GUI is flat, the edges of the GUI will be further away than the center, but if the GUI is curved, the distance can be the same in the center and edges.
6. **Provide visual feedback on interactive elements.** The interactive elements include elements in GUI and the objects in VR world.
7. **Use wearable menus.** This suggests that using a smaller menu worn by the user is a better approach. The menus that appear in the VR world might take a lot of space to occupy valuable space.
8. **Scale and spacing suitable for interactive elements.** Interactive elements need to be appropriately spaced to avoid accidentally doing unintended actions.
9. **Avoid pinning GUI in the user's view.** This can cause nausea or "cybersickness." If the UI needs to be pinned to the user's view, having the UI follow the user's perspective with a slight delay can be an option. This allows the user to look around before being obstructed by the UI.
10. **Avoid content in peripheral areas.** This is to reduce neck strain and improve the ergonomics of the game. Less important content can be put in the user's peripheral vision.
11. **Prevent the virtual hand from obscuring interactive elements.** If the user needs to pick up objects or interact with something, but the hand is obscuring the object, then the hand is making the task harder. Some technique to circumvent this is to have an object big enough to show around the hand even if the hand is holding the object. The hand can also be transparent, allowing the user to look through the hand.

Guidelines	Oculus	Unity	Leap Motion
Provide depth tip	X		
Comfortable content distance	0.5m to 1m		
Incorporate the GUI into the user's environment or character	X	X	X
Use texts in UI that are easily read	X	X	
GUI surrounding the user	X	X	
Provide visual feedback on interactive elements			X
Use wearable menus			X
Scale and spacing suitable for interactive elements			X
Avoid pinning GUI in the user's view		X	X
Avoid content in peripheral areas			X
Prevent the virtual hand from obscuring interactive elements			X

Table 3.1: *This table very close to a table from the paper[26]*

The guidelines were tested against the top five VR games on steam in July 2018. This list includes *Beatsaber*, *GORN*, *Hot Dogs, Horseshoes & Hand Granades*, *Rec Room*, and *Waltz of the Wizard*. The paper found that most of the games followed the guidelines. The least followed guideline was the use of wearable menus. The paper's authors did not have VR equipment and used videos to examine if the games followed the guidelines. The paper also conducted an online survey targeting VR players. The survey focused on VR UIs and found that the text often was pixelated, small, blurred, or in peripheral vision. This leads to the text being hard to read. Immersion was also a problem with the UIs. The UIs broke the immersion. They also pointed out that UIs are often confusing to access and control. The respondents had some solutions for UIs. Some of the suggestions included creating a more natural interface with gestures and making menus more straightforward. The problem with UI fixed to the user's view was another problem. Having the UI in front of the user obscured the environment. The respondents suggested that the UIs should be in a particular place and don't follow the user's vision. The respondents also recommended that the interactions in GUIs should be more physical. Respondents rated how important the different guidelines were. Most guidelines were important, but "Avoid pinning GUI in the user's view" had lower ratings than other guidelines. Implying that is a less important guideline. The authors of the paper did not have VR headsets and could not test the top games themselves. There could be more guidelines the top games followed but were not seen in the videos they reviewed. The guidelines' importance and the response from VR users are valuable to our paper as it helps give insight into how users see UIs.

3.6 Immersive Interface in Virtual Reality

In 2021 a master's thesis called "Immersive Interface in Virtual Reality" tested how traditional interfaces should be altered to create a more immersive experience in Virtual Reality[28]. The interface type was focused on the FPS(First Person Shooter) genre. The thesis created multiple prototypes with different interfaces. The different prototypes communicated how many bullets were in the gun's magazine. The different interfaces in the different prototypes were non-diegetic, diegetic, and spatial. The non-diegetic user interface displays the information on the screen as an overlay. This means that the information is always available. A diegetic user interface shows the information in the world, much like in the real world. A spatial User interface is an interface in the world that is only visible to the user.

The **non-diegetic user interface** used an overlay to display the number of bullets in the magazine. The number was placed in the lower right of the VR headset. This is a common way to show the info in FPS games on the computer.

The **diegetic user interface** added a window to the magazine. This allowed the user to count the number of bullets in the magazine but was not visible when the magazine was loaded into the gun.

The **spatial user interface** added a number next to the gun. This number indicated how many bullets were left in the magazine. The user interface only appeared when certain conditions were met.

Three types of immersion were tested: environmental, interactive, and realistic. The diegetic achieved the best scores in environmental and realistic immersion, and the non-diegetic achieved the best score in interactive immersion. The thesis claims that more experienced VR users might be more immersed when using diegetic or spatial user interfaces.

Key elements for deciding what type of user interface from the thesis:

1. Define your primary player.
2. Diegetic UI increases realistic and environmental immersion, but can severly affect interactive immersion.
3. If choosing Diegetic UI, extensive user testing is suggested to ensure that the difficulty does not become too high as a result of the lowered interactive immersion.
4. Spatial UI seem to be a good center-point between the traditional Non Die-

getic and Diegetic UI when providing information to the player.

5. If the Spatial UI can be hidden when not relevant, the perceived environmental and realistic immersion appear to be greater.
6. If using Spatial UI, beware of disrupting the gameplay.
7. If using Spatial UI, beware that the UI might be positioned behind other game objects at certain scenarios.

Chapter 4

Method

4.1 Research Method

4.1.1 Research Process

When conducting research, one must define all relevant aspects, such as the research topic and methods used to generate results. The idea of the research process is to take one from asking the research questions to answering them by presenting evidence that argues for whatever conclusion is drawn based on the literature review and results [29]. An overview model of the research process can be seen in figure 4.1. The model depicts various possible research strategies and methods for data generation and analysis. The red boxes in the figure denote the methods and strategies used for the research process in this thesis.

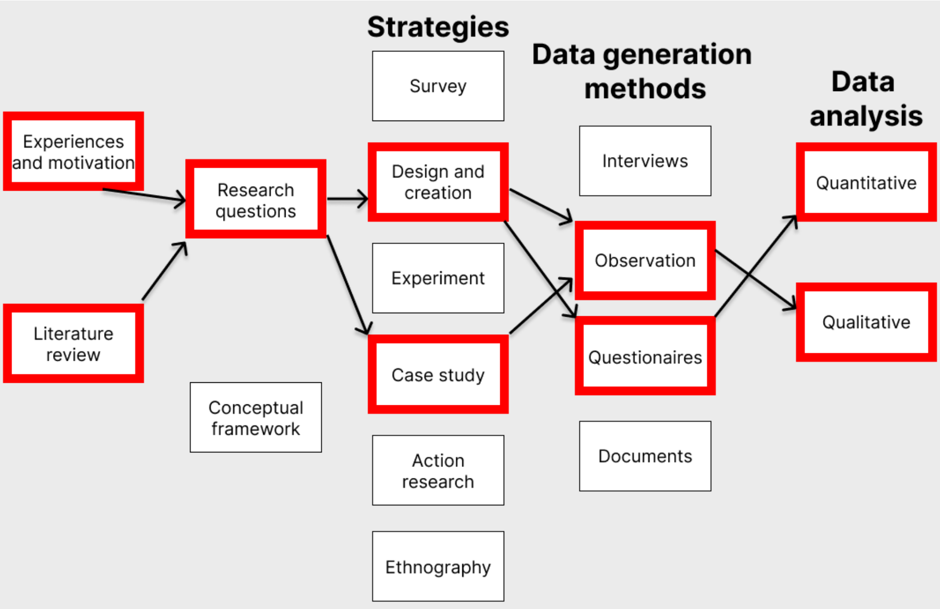


Figure 4.1: Recreation of the research model from (Oates 2006)[29]

In addition to the case study, design and creation were chosen as the research strategies for this thesis as the goal is to examine existing apps and design and create a new one. Data is collected in the form of questionnaires and observations. To analyse the data generated, qualitative and quantitative methods will be used.

4.1.2 Strategies/Design and Creation

The research was divided into two main parts for this particular thesis. The first is a case study to see what previous virtual internships had done regarding UI design. This involves testing them and evaluating their UI design. The second part is the main focus of this thesis, which is to create a new virtual internship app. The former is a case study, while the latter uses the design and creation strategy.

4.1.3 Case Study

A case study is a type of research where one focuses on a single instance of a thing [29]. This instance can be anything from individuals to entire organizations and information systems. The purpose of a case study is ultimately to gain detailed insight into the case being researched. This insight can be used to generate knowledge, which can be used in other types of research [29].

Case studies can be categorized into three basic types: exploratory, descriptive, and explanatory. For this project, the exploratory use case is chosen as one of the research methods. One of the characteristics of such a study is that there is little research on the topic, which necessitates the investigation of an instance to discover the topics to be covered in later research [29]. This fits the focus of this master thesis, as the goal is to explore ways to improve system usability in job taste apps. It is, therefore, necessary to study the previous job taste apps to see what they have done regarding usability to generate the knowledge needed when designing and creating the carpenter app. The case study details are explained further in chapter 5.

4.1.4 Design and Creation Strategy

The design and creation strategy is a strategy in which new IT products are created, often called for *artefacts*. In this strategy it is common to use an iterative process where 5 steps are followed: *awareness*, *suggestion*, *development*, *evaluation* and *conclusion* [29]. In the awareness step, the problems are identified and written down. Suggestion, as the name implies, suggests a solution to the problem defined in the previous step. The solution derived is usually not final as its main purpose is to explain how the problem might be solved. Development is where the suggested idea for the problem is implemented. The evaluation explores whether or not the product produced lives up to the expectations set and if it has value. In the conclusion step, discovering knowledge from the results is written down, and possible unexpected/inexplicable results lead to the foundation of future research. These steps, as mentioned, are followed in an iterative process. By doing it in a cycle, the five steps complement each other. The problem might not be understood correctly in the first iteration, which can become apparent when the suggested solution does not work when implemented. This blunder in the development step can lead to valuable insight into why things did not work, such as the suggested solution being not good enough or the problem description being flawed. This can lead to better problem articulation and new and better solutions that are then implemented. Therefore, the design and creation strategy is a methodology where knowledge is gained through the process of making a product [29].

4.2 Work Distribution

As there are two students from different master's programs, there are aspects in which one student excels slightly better than the other. One of the students studies computer science with a specialization in software systems, while the other studies Informatics with a specialization in interaction design, game, and learning technology. Both students studied the same bachelor's, and there are many similarities between informatics and computer science. This means a large amount of overlap in the competence between the students. Therefore, most of the products were developed by both students. However, there were areas of the application that was done entirely or primarily by one student. The similarities and differences in work distribution are highlighted in table 4.1.

Student(s)	**Task**
Both	Creating questionnaires and conducting user testing
Eivind	3D modelling in Blender
Aleksander	Creating task and language systems
Eivind	Creating both UIs
Aleksander	Creating circle saw and frame task
Eivind	Creating wall task and workbench task
Both	Fixing bugs
Eivind	Creating tutorial
Aleksander	Creating feedback and waypoint system

Table 4.1: Table showing a rough distribution of the work done.

As can be seen from the table, tasks were generally split between the students. This does not mean that one student created that task in its entirety singlehandedly, as pair programming was used intensively. That being said, some tasks were predominately developed by a single student, who made one student more responsible for specific tasks. The only job that one student did was 3D modeling in Blender. One of the students had never used this technology before, while the other had used it moderately. Since not many models needed to be made from scratch, it was decided that it would be best to let the experienced student be responsible for creating 3D models to utilize our combined resources better.

4.3 Data generation

This section explains the different methods used for gathering data. Both qualitative and quantitative methods will be used for data generation such as observations, questionnaires and any potential feedback that the users may have.

4.3.1 Observations

Observations involves the act of *watching* or *paying attention to* what someone is doing [29]. Observations can be done using either of the five human senses, but for this project, observation will entail the act of watching what users do when using a VR app. There are two types of observations – overt and covert observation. In this case, the former is used as the users testing the app are aware that they are being observed and that the observations will be used for academic purposes.

The advantage of overt observations is that the observer can interact with the user during testing. This makes it possible to ask the user questions about different things, such as things they thought were good or bad about the app and what they think could improve the app. By extension, it is also possible to ask the user to “think out loud” to find out what and how they think when using the product. This is especially important as it can highlight the discrepancies between the thought process of the user and the developer. Something that might appear evident to the app’s creator might not be self-evident for someone who has never tried it before. However, the downside of overt observations is that users know that they are being observed. The awareness of being observed can lead to a “Hawthorn Effect” in the user [29]. Such users will behave differently since they know their actions are being observed. They might not give genuine feedback, which could produce overly positive or negative data.

4.3.2 Questionnaires

Questionnaires are a data generation method where users are asked predefined questions in a particular order decided by the creator [29]. The purpose of a questionnaire is to gather quantitative data from users, which can be interpreted later to find trends and patterns. A questionnaire can either be self-administered or researcher-administered. For our purposes, users were given a self-administered questionnaire. In such a questionnaire, the user answers all the questions without the researcher(s) being present. This is beneficial for two reasons. Firstly, the researchers do not need to spend time with the users when answering questionnaires. Secondly, it means that the data gathered is anonymous as the researcher has no way of correlating the answers to users.

The questionnaires were made and conducted using *Nettskjema*. This is a website that allows researchers to create questionnaires quickly and efficiently and makes it possible to let users answer them digitally by using a phone or laptop. The questions created for the questionnaire in this project were based on existing ones, such as those made for existing Immersive Job Taste apps. In addition, system usability questions are needed when calculating a SUS score. Inquiries were made in Norwegian and English to cover a broader number of users. To let users answer the questionnaire, a QR code was printed, linked to the questionnaire hosted by Nettskjema.

4.3.3 Ethics

Since our research involves other people, it is important that ethics are considered and taken care of before commencing user testing. Most institutions require that researchers gets ethical clearance before they can start [29]. To ensure that the data we collect is done in an ethical manner, a consent document is prepared in advance before conducting user tests. The purpose of such a document is to inform the user about what kind of data that will be collected, for what purpose, how it will be stored, and things of that nature. With a consent document, the user can read the terms and make an informed decision about whether or not they want to participate in our research. If the user decides that the terms are agreeable, and wishes to participate in the user test, they are required to sign the consent document with a pen as proof that the data was collected in an ethical manner. The consent document used in this project was provided by our supervisor.

4.3.4 User tests

Iteration	Users	Count
First iteration	NAV employees	11
First iteration	VR4VET partners	11
Second iteration	Teachers	18
Third iteration	NAV employees	5
Third iteration	Young job seekers	5

Table 4.2: User types and number of users in various user tests conducted throughout the different iterations

4.4 Data Analysis

Data gathered can be classified as either *quantitative* or *qualitative*. Quantitative data is any data or evidence which can be quantified or counted. In other words, it is data based on numbers [29]. There are several types of quantitative data. However, for this thesis, nominal and ordinal data are collected. Nominal data is any data without numerical values such as categories (i.e., gender, occupation, address). Ordinal data includes numerical data that is not proportionate but where the order matters. For instance, the Likert scale is an example of ordinal data as it uses the numbers 1 through 5 to measure how much someone agrees with a statement. However, it is impossible to say how much difference there is between a Likert score of 4 and 5, only that a score of 5 is *better* than a score of 4.

Qualitative data is all data that is not numeric, such as words, images, sounds, etc. [29]. Qualitative data collected for this project will mainly be observations done on users and any feedback they provide.

Chapter 5

Case Study of Earlier Applications

This chapter contains the full details of the case study done on earlier job taste apps. There are two parts to this case study. Since the aim is to investigate usability in the previous job taste apps, it is paramount that guidelines for good UI are established. Therefore, the first part consists of the guidelines found during this literature review, while the second looks at how these are enforced in the VR apps.

5.1 Guidelines for Creating Good UI in VR

One of the main components of creating a good user experience in any VR game is to have a good UI. This is an important aspect as UIs show up in every job taste app created to date. These relay information and can often be a point of interaction for the user. Thus, they must be well made to ensure a good user experience.

In the first phase of the project, a thorough literature review was done regarding the act of creating good UIs in VR games. Fortunately, such guidelines are already defined in documents produced by various big companies that focus on VR games. Since there are plenty of guidelines from different companies, it becomes quite a long list, and not every guideline made by these companies is mutually agreed upon. To simplify things, inspiration was taken from earlier research presented in chapter 3.5. This paper compares a couple of guidelines made by different sources, Unity, Oculus, and Leap Motion. These results are comprised in a table showing each corresponding guideline and which of the three sources have this included in their guidelines. These guidelines have been used when evaluating the UI of the previous job taste apps, and the findings can be found in table 5.3.6.

5.2 Testing Earlier VR Apps

One of our goals is to see what can be done to create a VR app that has a good user interface and is user-friendly and thus easy to use. A lot of inspiration was drawn by testing the previous Virtual Internship apps that had been made to see what they did regarding playability and UI. During this period, the following apps were tested:

- FiskeVR / FishVR
- Blikkenslager / Tinsmith
- Bilmekaniker / Car mechanic
- Vindmølleteknikker / Windmill turbine engineer
- Veiarbeider / Road construction worker

It should be noted that no user testing was conducted during the case study. Instead, the master's students would test these 5 VR apps. During testing, the UI of the apps would be evaluated using the predefined guidelines mentioned in 5.1. In the beginning, the focus of the master thesis was to look at how to improve the UI in job taste apps, which is why no guidelines for playability are presented. Despite this, playability issues or boons would still be noted when testing to figure out what pitfalls to avoid and what to follow when creating the carpenter app. As a result, two kinds of results will be presented for each app – the guidelines followed and any comments regarding playability.

The job taste apps were tested on IMTEL's VR lab at NTNU Dragvoll. Depending on the application, the apps were tested using either the HTC Vive or the Oculus Quest. Since both headsets were connected, it became app dependent on which headset was used. All the apps mentioned above are already installed into a program called "Yrkeskatalog," which roughly translates to "Profession catalog." As the name implies, it contains a catalog of different job taste apps from previous students. In this catalog, more than the five apps were tested in this case study, but for various reasons, not all of them could be played. For instance, the crane app would not work with either VR headset, and the warehouse app is a desktop app and not a VR app. Therefore, only the apps that worked and had a VR version were tested. During testing, one student used the app through the VR headset while the other wrote down the things that were observed. This includes going through all the established guidelines and other observations regarding the app's playability. This ranged from items deemed as being not user-friendly to things that positively affected the user experience.

5.3 Results

Here are the results of testing the five different job taste apps. Each app will have its section highlighting the guidelines that were followed as well as any observations that were made when it comes to general playability. For the guidelines section, there will be a table for each app displaying whether or not the app followed the guidelines. For each cell in this table there are 3 possible results, either *Yes*, *No* or *Partially*. The yes and no options state whether or not the guideline was entirely fulfilled. Partially, on the other hand, means that the guideline was followed to some extent meaning that one part of the app followed it while another did not. Lastly, there are some general comments on the user-friendlinessliness of certain gameplay aspects.

5.3.1 FishVR

UI guidelines

Guidelines	**Fulfillment**
Comfortable content distance 0,5m -1m	Partially
Incorporate the GUI into the user's environment or character	Yes
Use texts in UI that are easily read	Yes
GUI surrounding the user	No
Provide visual feedback on interactive elements	Yes
Use wearable menus	Partially
Scale and spacing suitable for interactive elements	Partially
Avoid pinning GUI in the user's view	No
Avoid content in peripheral areas	No
Prevent the virtual hand from obscuring interactive elements	Yes

FishVR tended to break many UI guidelines either entirely or partially. It had a comfortable content distance, for the most part. However, in the beginning, the text would be close depending on where the user would stand. If the user had more freedom to move, it would be easier to adjust for this. Still, there are only a handful of predefined teleport locations, meaning the text placement should ideally be moved back for a more comfortable reading experience. Besides this, the text is usually placed at a comfortable distance and is easy to read.

The placement of the GUI is mixed. Most text is placed statically in the game world and does not move. Menus, however, follows the player in one shape or form. The tablet that contains info about tasks is pinned to the user's view, which is why this app did not follow this guideline. There is a second menu that the user can open, which follows the controller/hand of the user. This means that FishVR has an element of wearable menus, but it is only for one of the menus. It also does not avoid content in peripheral areas, as in the beginning, the player is placed underwater, surrounded by three big walls of text. When looking at one of the walls, text from another wall bleeds into the peripheral area of the user.

The spacing on interactive elements was followed partially. This was not a problem on the tablet but in the wearable menu. Some buttons are too close together and could benefit from being spaced out more. Another instance of this was in the feeding station. When feeding the fish, numerous buttons can be pressed. Some choose which fish location to provide, while the other buttons control the feeding itself. There is visual feedback on interactive elements in this app. Still, for the feeding station, the controller is highlighted and not the button selected, making it challenging to know which button was being pressed.

Observations

- Some texts have unfavorable background colors, such as the text at the beginning having white text on a background with bright color.
- There are a few bugs in the app, such as the tablet being able to teleport the user to arbitrary locations. This happened when leaving the feed station and attempting to use the boat. The tablet was activated when trying to start the boat, and the user was teleported back to the feed station. There was another bug where it was impossible to teleport from the boat to the fishing cages.
- The wearable menu tended to reduce the framerate significantly. There was also no feedback when pressing buttons on this menu.
- The app had a small tutorial that showed how to teleport. This seems handy since it highlights which buttons to press.
- The welcome audio message on one of the stations would frequently loop, which became very annoying.
- The feeding station should have a better explanation of exactly what to do and give the user some feedback when feeding the fish.

5.3.2 Tinsmith

UI guidelines

Guidelines	**Fulfillment**
Comfortable content distance 0,5m -1m	Yes
Incorporate the GUI into the user's environment or character	Yes
Use texts in UI that are easily read	Partially
GUI surrounding the user	No
Provide visual feedback on interactive elements	Partially
Use wearable menus	Partially
Scale and spacing suitable for interactive elements	Partially
Avoid pinning GUI in the user's view	Yes
Avoid content in peripheral areas	Yes
Prevent the virtual hand from obscuring interactive elements	Yes

The tinsmith app did slightly better in the UI department as fewer guidelines were broken. The text was always distanced well and almost always readable. The only exception is text on the tablet. For the text on the tablet to be readable, the user has to stretch their arms out so that the tablet is positioned optimally for viewing pleasure.

The placement of UI was well done in this app. It is either incorporated into the environment, such as floating text or a wearable menu that is not pinned to the user's view. It also avoids putting content in the peripheral areas of the user.

While the spacing in the tablet is done well, the spacing for the cutting task could be improved. At the cutting station, there are buttons for navigating through different metal shapes that need to be cut. These are placed a little too close together, leading to pressing the wrong button sometimes.

Visual feedback is prominent in the app for the most part, except for the tablet. When hovering over buttons and pressing them, there is no visual feedback to indicate that the button has been pressed. On the flip side, the objects that can be grabbed and machinery that can be interacted with have outlines to show that they can be used in some way.

Observations

- There was no explanation on how to use the pliers on the cutting task.

- Using some tools like the pliers was complicated, especially on the HTC Vive. This tool, in particular, did not work as expected. The initial thought was that the blades of the pliers had to come into contact with the metal plate to cut. Instead, the user must utilize the laser that shoots out from the pliers to guide where to cut.
- Some of the tasks were hard to do, especially the cutting and bending tasks.

5.3.3 Car Mechanic

UI guidelines

Guidelines	**Fulfillment**
Comfortable content distance 0,5m -1m	Yes
Incorporate the GUI into the user's environment or character	Partially
Use texts in UI that are easily read	Yes
GUI surrounding the user	No
Provide visual feedback on interactive elements	Yes
Use wearable menus	No
Scale and spacing suitable for interactive elements	Yes
Avoid pinning GUI in the user's view	No
Avoid content in peripheral areas	Yes
Prevent the virtual hand from obscuring interactive elements	Partially

The car mechanic app had sufficient distance for the text displayed and was readable wherever it occurred. This is another app that incorporated the tablet, and similarly to FishVR, it is pinned to the user's view. The only UI found was a board with text and a diagram showing how to switch relays in a car. Because of this, the app partially incorporates the GUI into the user's environment.

The tablet has good spacing for its buttons, but the text could be spread out more. The tablet contains the different tasks and the number of points achieved for the respective task. This list of points is texted bunched together with no space between them, which can be a bit daunting to read. Finally, the app does prevent the virtual hand from obscuring interactive elements for the most part. They were slightly in the way when handling the bolts when changing tires, although this might be inevitable as the bolts are small relative to the hands. The hands were not obscuring in any significant way.

Observations

- It had auditory and visual feedback, which gives a good indication if something has been done correctly.
- The 360 videos are a good tool for showing how something is done, but it tends to restart if the user moves too far away.
- The different tasks have their page on the tablet for keeping track of the score. This gives a good overview of the tasks and how well they were executed.
- When changing the lights on the car, it would be ideal with a bit more explanation. The 360 videos did not convey how to do the task correctly.
- Some of the bolts on the wheel task were not very visible.

5.3.4 Windmill Turbine Electrician

UI guidelines

Guidelines	**Fulfillment**
Comfortable content distance 0,5m -1m	Yes
Incorporate the GUI into the user's environment or character	Yes
Use texts in UI that are easily read	Yes
GUI surrounding the user	No
Provide visual feedback on interactive elements	Yes
Use wearable menus	No
Scale and spacing suitable for interactive elements	Yes
Avoid pinning GUI in the user's view	Partially
Avoid content in peripheral areas	Yes
Prevent the virtual hand from obscuring interactive elements	Yes

While it might not be apparent from the table above, the windmill app is perhaps the best app for following the guidelines. The text is always readable any time it appears. GUI is placed very nicely in almost all instances. The menu is locked in world space, and it is larger than the tablet making it easy to navigate and read. That being said, there is one instance of the GUI being pinned to the user's view: when they have completed a task. After completing a task, a tooltip pops up to inform the user that the task has been completed. This tooltip is pinned to the user's view and follows the headset. Despite this, it is only up on the screen for a few seconds and does not block the user's vision in any meaningful way.

Observations

- It was nice that there exists a tutorial on how to teleport. It would be helpful to have a tutorial for other things as well.
- Navigation was done well. There are markers to show where to go, and it is possible to teleport to a handful of chosen destinations. This is a feature that works well on pretty big maps.
- It was a bit weird that the user has to spawn in lightbulbs from the menu when changing the lightbulb at the top of the windmill.
- Having 3D models of VR controllers instead of hands seems like a good idea since it makes it possible for the user to see the different buttons on the controller without having to take off the headset.
- A lot of the tasks were self-explanatory, especially with floating text giving helpful information.
- Overall straightforward to use, and few bugs were found.

5.3.5 Road Construction Worker

UI guidelines

Guidelines	**Fulfillment**
Comfortable content distance 0,5m -1m	Yes
Incorporate the GUI into the user's environment or character	Yes
Use texts in UI that are easily read	Yes
GUI surrounding the user	No
Provide visual feedback on interactive elements	Partially
Use wearable menus	No
Scale and spacing suitable for interactive elements	Yes
Avoid pinning GUI in the user's view	Yes
Avoid content in peripheral areas	Yes
Prevent the virtual hand from obscuring interactive elements	Yes

The road construction app also did not deviate very much from the guidelines. Among the five apps, this was the only one that did not have an interactable menu. Instead, it had a board stationed in the middle of the map, which displayed the user's score on the different tasks. The text here was easy to read, and the spacing was good. All UI and text found were stationary in the game world, and content would not enter the user's peripheral area. There was some visual feedback when doing actions in the game such as using the shovel to dig dirt or connecting pipes. The pipes had a highlighted area that showed the user where to place the pipe.

Despite this, there was no visual feedback for objects that could be grabbed. This made it difficult to know if an object in the game could be picked up.

Observations

- It was a bit nauseating to drive the excavator.
- The pipe task had highlighting and snapping, which made it easy to both understand what to do but also to perform the task.
- The excavator had unexpected controls. Only one stick can move in 2 directions, and it is both used for rotating and driving the excavator. This control mapping felt weird and was a bit difficult to use, but it might be how an excavator works in real life.

5.3.6 Comparison

Guidelines	**FishVR**	**Tinsmith**	**Car mechanic**	**WindmillVR**	**Road construction**
Comfortable content distance 0,5m -1m	Partially	Yes	Yes	Yes	Yes
Incorporate the GUI into the user's environment or character	Yes	Yes	Partially	Yes	Yes
Use texts in UI that are easily read	Yes	Partially	Yes	Yes	Yes
GUI surrounding the user	No	No	No	No	No
Provide visual feedback on interactive elements	Yes	Partially	Yes	Yes	Partially
Use wearable menus	Partially	Partially	No	No	No
Scale and spacing suitable for interactive elements	Partially	Partially	Yes	Yes	Yes
Avoid pinning GUI in the user's view	No	Yes	No	Partially	Yes

Avoid content in peripheral areas	No	Yes	Yes	Yes	Yes
Prevent the virtual hand from obscuring interactive elements	Yes	Yes	Partially	Yes	Yes

Based on the results, it is clear that most apps either followed the UI guidelines or did so to some extent. There was one guideline that no app followed which was having a GUI that surrounded the user. This is arguably not that important of a guideline as the text was readable most of the time, and the menus were not challenging to navigate. Most apps did also not have a wearable menu, or it was only partially. About half of the apps did not avoid pinning the GUI in the user's view, but everyone incorporated it into the user's environment or character. This seems to be an important guideline as it is present in Unity, Oculus, and Leap Motion's guidelines. The text was something every app seemed to do well when it comes to readability, with few exceptions, namely FishVR and tinsmith, who had some unusual quirks that made reading difficult in specific instances. Generally, the apps have rather well-made UIs, although there is room for improvement.

Playability, however, had a more significant discrepancy between the apps. Some apps like the car mechanic or tinsmith were challenging to use due to poor explanations. Despite having 360 videos, it was not always enough information to understand what needed to be done. It also does not help when controls are not fully explained, making it sometimes nonintuitive to perform the actions required to complete the task. The windmill app, on the contrary, was straightforward to use. In addition to 360 videos, a lot of text was scattered everywhere, giving valuable information. The controls are also explained well by having a 3D model showing all the buttons.

5.4 Requirements

Based on the findings in the case study, some requirements for developing the carpenter app were made:

- Create a new UI following the guidelines:
    1. Comfortable content distance 0,5m -1m
    2. Incorporate the GUI into the user's environment or character
    3. Use texts in UI that are easily read
    4. GUI surrounding the user
    5. Provide visual feedback on interactive elements
    6. Scale and spacing suitable for interactive elements

    7. Avoid pinning GUI in the user's view
    8. Avoid content in peripheral areas

- Implement the UI into the application with navigation and task descriptions.
- Create tasks based on using a nail gun
- Create task based on sawing
- Follow the guideline: Prevent the virtual hand from obscuring interactive elements
- Create a tutorial

Most requirements are straightforward as they are the guidelines. The only guideline not followed was to use wearable menus. This was decided against as it would be hard to create and design in such a way that the users could easily interact with it. The guideline to prevent the hands/controllers from obscuring interactive elements is important. This guideline was thought easy for the carpenter app to follow because most of the interactive objects would be larger than the hands/-controller.

Chapter 6

Design and Creation of CarpenterVR

6.1 Software Tools

6.1.1 Unity

Unity is a development platform, with over 50% of the world's videogames made by using unity as the development platform[30]. Some of the industries that use Unity include the architecture, automotive, film, and the game industry[31][32][33]. Unity has the possibilities to create 2D, 3D, AR, and VR experiences[30]. Using Unity as a development platform has multiple positive features. It has a large asset store with community backing, a physics engine, and an advanced graphics rendering pipeline, and it is free to use. Unity lets the developer use C# to make scripts. Addons or frameworks are easily added to the project thru the Unity package manager.

6.1.2 VRTK

VRTK stands for Virtual Reality Toolkit and is a collection of reusable solutions to common problems when creating VR experiences[34]. This toolkit helps speed up the creation process as VRTK has 34 different packages that help development[35]. Some tools directly interact with the application. Others allow the developer to test without having a VR headset connected. This is called a "Spatial Simulator" [36]. This "Spatial Simulator" lets the developer move the body and hands. It also enables the developer to bind different buttons from the VR controllers to the mouse and keyboard. This speeds up testing significantly, but testing with an actual VR headset is necessary before testing on users. VRTK is released under the MIT License and is completely free for anyone to use.

6.1.3 Blender

Blender is a free and open-source program for 3D creation with a suite of tools. Blender supports the entire 3D pipeline: modeling, rigging, animation, simulation, rendering, compositing, motion tracking, video edition, and game creation[37]. Blender is cross-platform and runs on Windows, Macintosh, and Linux. Blender can export to the .fbx file type, which Unity supports.

The main use of Blender in the development process was:

- **Creating 3D models for the application.** The process of creating 3D models is to create a mathematical coordinate of the surface of an object. This can be done by manually typing a list of vertices and a list of polygons, but it is much faster to do with software like Blender.
- **UV-unwrap the 3D models.** UV-unwrapping a 3D model is taking the 3D object and unwrapping it to a 2D plane. This is done because the texture is 2D and not 3D.
- **Creating animations with 3D models.** Animations are made by having one or multiple objects change scale, position, and/or rotation over time. The change is saved as multiple keyframes with the scale, position, and rotation, plus the timestamp. The animation moves from one keyframe to another gradually.
- **Applying textures that were already created.** To apply textures, the object needs to be UV-unwrapped. The UV-unwrapping is the most crucial step as the object with incorrect UV-unwrapping might not look correct.
- **Creating materials for 3D models and "baking" them to a texture map.** Blender can create textures for any object. The process is a bit complicated, but it is a combination of nodes with different properties, as seen in Figure 6.3. Blender has over 80 different nodes; the more complicated the object's material/texture, the more extensive the node network. When a material is made this way, it must be "baked." "Baking" is the process of rendering the material to a texture map. The reason for doing "baking" is for programs other than Blender to use the textures that have been created.

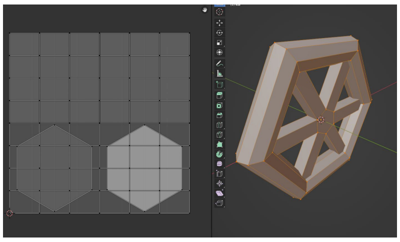


Figure 6.1: Image of a 3D object to the right and the same object UV-unwrapped to the left

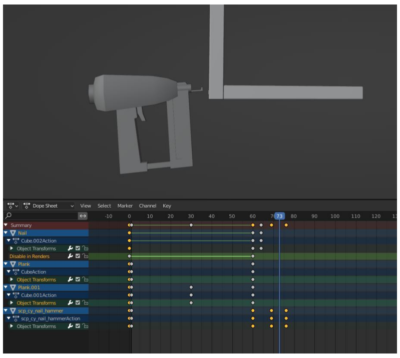


Figure 6.2: A animation of a nailgun. The keyframes under contain each objects scale, position and/or rotation

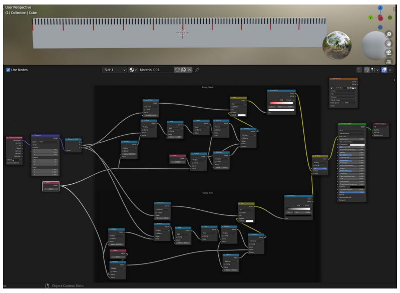


Figure 6.3: The creation of a ruler texture on a object. The nodes at the bottom of the image and the object with the texture at the top

6.1.4 Figma

Figma is an online UI design tool that combines the accessibility of the web with the functionality of a native app[38]. This tool lets users easily collaborate and share in the design process. Figma has features that help save time. One of these features is components. Components consist of one or multiple items like text, lines, boxes, and circles. Copies of components point to the original component, making them mirror the original component. This makes copies of the main component change depending on the changes applied to the main component. This makes it easy to make changes to the design, like changing colors or font. Creating a UI in Figma and importing it to Unity is somewhat easy. Figma can export components either in the PNG file format or as CSS code. This is useful as using PNGs in Unity is an easy way to create UI elements.

6.1.5 Git

Git is a valuable tool for collaborating. It lets multiple developers make changes to the application and saves them in the cloud. This makes collaboration easy as multiple developers can make changes to the application simultaneously and combine them when necessary. Git also tracks these changes and saves them in a way that lets developers go back to earlier versions of the application.

6.1.6 Trello

Trello is an online web application used to keep track of tasks. It is a visual tool that lets users create a virtual Kanban-type board. Tasks are saved as cards and placed in lists. These lists have different titles that describe the cards. For example, a list with the title “To do” represents the cards that have not yet been started. These cards can be freely moved between lists to update the task’s status. This is how different tasks were distributed between iterations.

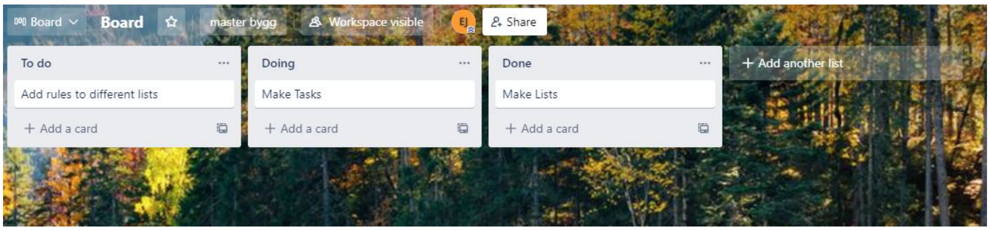


Figure 6.4: Trello with lists and cards

6.2 Hardware

6.2.1 Oculus Quest

The Oculus Quest is an all-in-one gaming system built for virtual reality. This means that the headset does not require a PC or similar device to play games, as it runs on the headset itself. The headset comes with two touch controllers, which can transport hands and gestures into the game being played. Using this headset without controllers is also possible as it supports hand tracking. With this headset, there are two main modes, stationary and room-scale. The stationary mode works by picking out a spot in the room where the player wants to play from. When a location is assigned, the player must stay within that area when using a VR application. The room-scale mode lets the user define a playable area by drawing borders around the playable space in the room. This enables the user to move around more freely as opposed to the stationary mode [39].

Chapter 7

First Iteration

In this chapter, the first iteration of the development cycle will be described. This includes the requirements for the job taste application, the features implemented, and the results of the user testing done on the prototype at the end of the iteration.

7.1 Requirements

The goal of this master thesis is to explore possibilities to improve the general usability in Immersive Job Taste apps. This includes gameplay aspects as well as UI. Based on the literature review and especially the case study, it became apparent that some requirements should be established when designing and creating the carpenter app. All of the initial requirements can be seen in table 7.1. These requirements were set before designing and creating the carpenter app.

ID	**Requirement**
1	The application has to be compatible with the Oculus Quest HMDs
2	Tasks should be added where the user can cut materials and nail together items
3	A tutorial needs to be added in order to explain the game
4	The application needs a UI that follows the guidelines found in the literature review
5	The application needs to provide scores in order to evaluate the user's performance
6	The application should give feedback to the player when they do tasks to guide them
7	The application needs to have a sufficient FPS preferably 90 or more
8	The application should use VRTK for quick development
9	The tutorial needs animations or "ghosts" to show what to do
10	The tutorial needs to teach users concepts stepwise

11	The application needs to be simple so that all users can complete the tasks

Table 7.1: The initial requirements set for Carpenter VR.

7.2 Setting up Development Environment

7.2.1 Version control - Gitlab

When developing the Immersive Job Taste app, it was essential to set up version control. There are several reasons for this, but for this particular project, it was paramount because two students were working on developing the VR app. Having version control is an excellent strategy for developing software when several people are involved. It was decided that GitLab would be used for version control. There are other alternatives, but the main reason for using GitLab was because all the earlier job taste apps are already on GitLab, so it felt natural to follow suit.

7.3 Features

The first iteration is by far the biggest because it had the most features implemented. This was necessary as this would become the prototype of the application that users would test. In the following chapters, these features will be described in detail.

7.3.1 Movement

The first thing implemented was a way for the users to move around in the virtual environment using the VR headset and controllers. For this, the VRTK SDK was used since it has packages containing VR rigs that can be added to the game scene in Unity. This rig allows us to see the game world and look around when equipping an HMD. Another VRTK package can be installed to make the controllers work, allowing us to utilize the VR controllers by mapping their buttons to perform specific actions within the game.

Initially, it seemed like a good idea to let the users choose how they would like to move by either walking or teleporting using an arc. The teleport part was relatively easy to implement since VRTK has a package that can be installed and added to the game scene. Teleportation was executed by making the joystick on the controller dictate the location in which to teleport, and by letting go of the stick, the player would be teleported to the marked area indicated by a green arc (see figure 7.1).

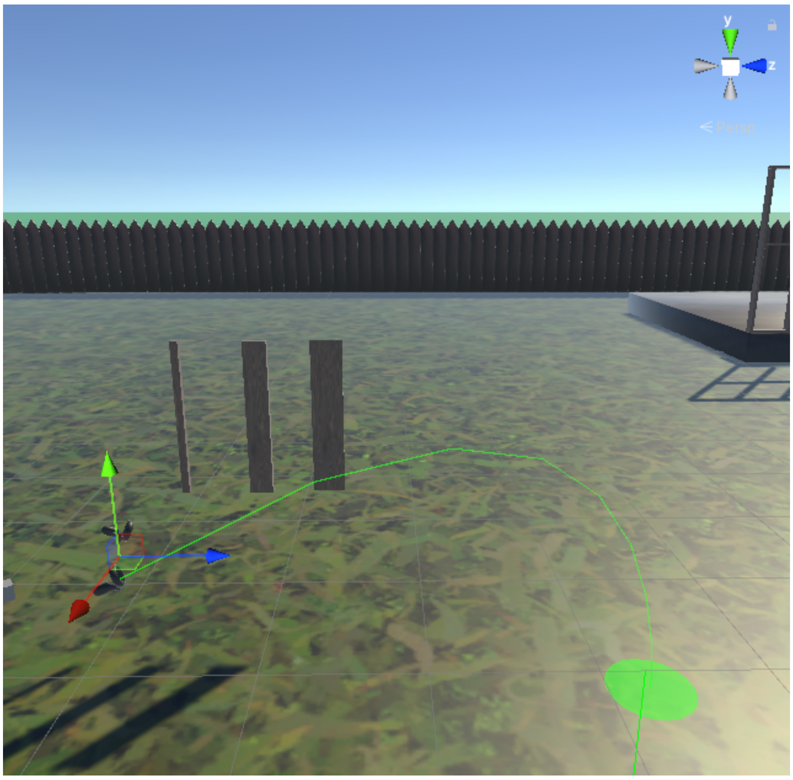


Figure 7.1: Image showing the teleport arc

Walking was made by reusing a script from the specialization project. Since camera rigs from VRTK were now used, minor adjustments in the movement script had to be made to make it compatible. With this movement script, users can walk around the scene by moving the joystick on the controller. These two movement systems were combined by adding each movement system to their respective controller. In this case, walking was done by moving the joystick on the left controller, while teleportation was done by moving the stick on the suitable controller.

7.3.2 Tools

After creating the movement system and the map, some commonly used tools in the carpenter profession were implemented. Typical work tasks in carpentry involve cutting and nailing wood to create something. The tools developed in this iteration were based on these two main tasks.

Circle Saw

The first thing was implemented as a way for the player to cut planks, as this is a typical work task for carpenters. Based on the research on the profession, it seems typical for carpenters to use circle saws to cut wood. A semi-functioning circle saw had already been made from the specialization project. However, some changes had to be made to the saw to work correctly. The circle saw has three main parts: the table, a cutting block, and the saw itself. The table is where the plank is placed when cutting, and the cutting block is used for guiding the plank towards the saw. A wheel that can spin in either direction was created to move the cutting block. Spinning the wheel causes the block to move alongside the plank it comes into contact with, making it possible to place the plank precisely, which is essential when it has to be cut to a specific length. The saw itself consists of a saw blade inside a protective shield that can be moved with a top handle. The saw blade has a predefined hitbox that detects whether or not a plank has come into contact with the saw. Once the plank leaves the blade hitbox, it will cause the plank to be cut in two.

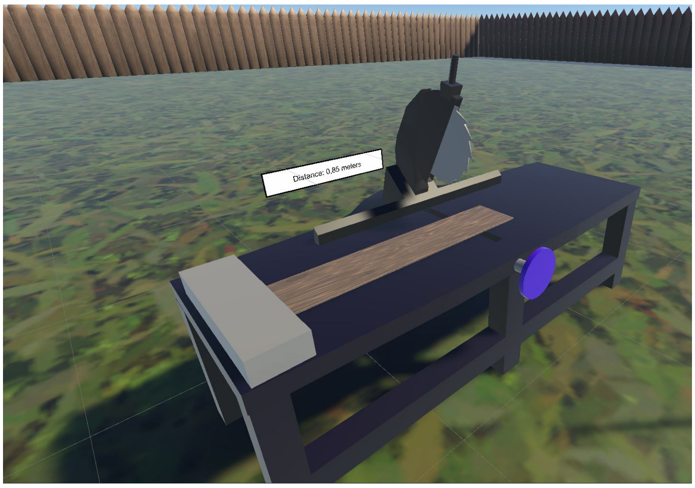


Figure 7.2: Image of the circle saw

Nail Gun

Much like the first tool, the second tool was imported from our specialization project. Again, changes had to be made since it was not functioning as intended. The nail gun is currently the only tool that can be picked up, which means it has to be grabbable. This was done using VRTK as it has its package for making things interactable, including grabbing. To make the nail gun grabbable, it was placed inside an interactable prefab object from the VRTK Interactable package[40]. To make the nail gun shoot nails, the shooting script it came with was reused. Changes were done to the hand to make it work with the new control system from VRTK, and some optimizations were done on the nails spawned to increase game performance. It currently works by having a hitbox that detects if the gun's tip is close enough to the object being nailed. If the distance is right and the object is nailable (i.e., a plank or similar object), a nail is spawned and attached to the object. If the nail hits several objects, these will become nailed together and work as a single object.

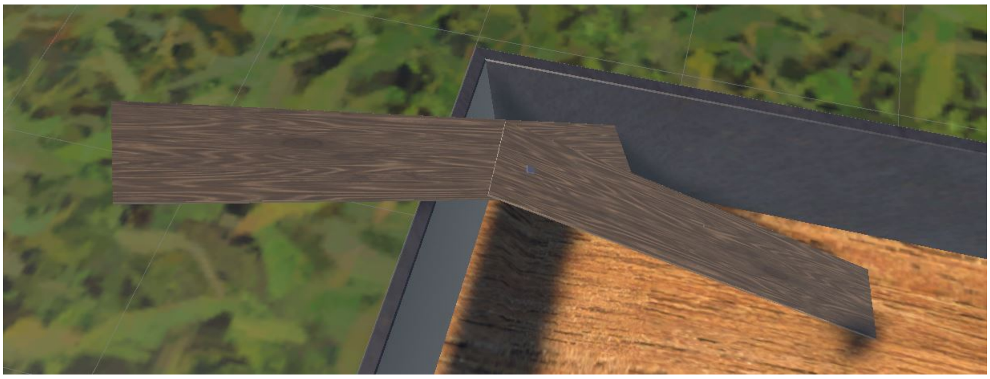


Figure 7.3: Two plank objects combined by a nail

Toolbelt

A toolbelt was added to make the game more realistic and improve the user experience in the form of convenience. The idea was that it would be a belt attached to the player where tools could be dropped and stored when they were not in use. The toolbelt consists of 2 slots where tools can be dropped. These slots were, for the time being, two grey cubes since no 3D models for these yet existed. For the tool storing logic, functionality from the VRTK packages was used again[35]. There exists a package with a prefab called a *SnapZone*[41]. These snap zones work by dropping things into them. This causes the item dropped in the snap zone to snap in place to a designated snap location. Once an object is snapped in place, it will be fixed in space until the player picks up the object again. 2 snap zones were placed inside the grey cubes on the toolbelt. The radius of the hitbox was set down to an appropriate level so that it would catch anything that the user dropped in the proximity of the belt. However, a rule was added to the toolbelt to avoid players dropping any object into the toolbelt so that the only objects it accepts are tools. This was done using the Rule system that Zinnia[42] has, which allows for all sorts of complex rules to be made. In this case, a rule is simply a condition that must be met for the snap zone to accept the item dropped in its radius. One of these rules is Tag Rules, which was added to the toolbelt so that it only accepts objects with the *tool* tag.

7.3.3 Trashcan

Since a carpenter often handles different materials such as wood and metal, it seemed like a good idea to have the means to dispose of them. Therefore, a trashcan was added where players could dump unwanted materials and other discarded objects. The trashcan 3D model was imported from a package on the Unity Asset Store. The logic for deleting items was then added to this 3D model. Currently, the trashcan has a hitbox that detects when an object enters it. If the object is a plank or similar item, the trashcan will delete the object from the game scene.

7.3.4 Materials

One of the most common materials that carpenters operate with is wood. Thus, different types of planks were added so that users could create a variety of things. 4 different types of planks were made for this iteration, 48mmx48mm, 48mmx148mm, 1-meter plank, and a 2-meter exterior plank. The difference between these types of planks is their dimension. The 48x48, for instance, is 48mm in width and height, while the 48x148 is 48mm in size and 148mm in width. The meter planks, surmised from their names, have different lengths. The 2-meter plank is also notable in that it is not a perfect rectangle, as is it an exterior plank which are planks used for building house walls. These planks were modeled in Blender and were given a wood-like texture. In addition, they have given a script called *Sliceable*. This script makes it possible for these planks to be cut using the circle saw. The planks are also given a tag called *Nailable*. This tag makes it possible for the nail gun to differentiate between objects that can be nailed and those that cannot.

7.3.5 Tasks

After the tools were implemented and tested, some tasks were made, so the player had something specific to do. In total, three tasks were made, 1 for cutting and 2 for nailing.

Task Manager

A task managing system was needed to create new tasks and keep track of them. In the beginning, there was a discussion on whether or not the task manager from the tablet provided by IMTEL should be used as it has seen prominent use in existing job taste apps. However, this system is designed to work with their tablet, which for this project was abandoned to explore how UI affected the usability of the application. A new system could be integrated with the UI (see section 7.3.6) was needed. This raised a dilemma: either transform the task manager system from the tablet to be compatible with the new UI or make a completely new system. Ultimately it was decided that creating a new system was the better idea since it would take time to become familiar with the previous system, which appeared to be unnecessarily complex for our use case.

In the new system, there are tasks and task managers. Each task manager is responsible for their respective type of task. For instance, the cutting manager is responsible for managing tasks that involve cutting materials such as planks. Likewise, the nail manager is accountable for tasks that consist in using the nail gun. Each task manager has a list of all the corresponding tasks. Each task has metadata such as the task's name, a description, how many points can be awarded, and whether or not the task has been completed. This task managing system is integrated into the new UI system described in section 7.3.6.

Cutting Planks

The first type of task that was implemented was cutting planks. The cutting manager manages these tasks, which involve cutting a specific plank type down to a certain length. It is expected that carpenters can cut materials to the correct length to be used for construction. A cutting task has two main things it looks for when evaluating if the user managed to perform the task correctly. The first part is to check if they are cutting the correct type of plank. If a player cuts a 48x48 plank to half a meter when the task explicitly says to cut a 48x148, the player will not be awarded any points. The second thing to check is that they have cut the length of the plank to what is specified in the task. For this system, no points are awarded to the player if they cut the wrong plank. However, they might get some points depending on how close the cut length is to the target length. It does a pretty basic check by seeing if the cut length is within 5% of the plank's original length before it was cut. For example, if the task is to cut a 1-meter plank down to 0.5 meters, and the user has cut the plank to 0.54 meters, they will still get some points for their work. Currently, the point system is very simplistic in that the awarded points are defined by a linear function that looks at the cut error as a proportion of the maximum allowed error. In this case, since the permitted error is 5% of the length of the plank (5cm for a 1-meter plank), and the player has cut with an error of 4cm (54cm – 50cm), the awarded points are 20% of the max points available for the task (1 – 4cm/5cm = 0.2 = 20%). This means that the smaller the error or disparity in cut length to the target length, the more points are awarded. Ideally, this function should be a bit more complex, since currently, it is almost impossible to get the maximum number of points possible since the measurement has a small margin of error, which leads to a lot of scores in the high 90s, which ideally should just be rounded up to 100.

Nailing Planks

Another common task that a carpenter does is to nail planks together. Since there was not much time at this point in the iteration, it was decided that the nail tasks would be very rudimentary. The two nail tasks added were: nailing a certain number of planks together and nailing exterior planks on a house wall. Both of these tasks are the same principle: to nail together X items, where X is the number of items specified in the specific task. The nail task manager will check if a task has been completed whenever two different objects are nailed together. It does this by checking if the nailed object contains exactly X number of objects that are not nails. If it has less than x number of objects, it will give the player a portion of the points and update the feedback message giving a status report on how far the player has progressed with the task.

7.3.6 Game Menu

A new game menu was created since UI and user experience is the primary focus. Some important guidelines were to have the proper distance between the player and the UI and to make the UI fixed in world space and not to the user's view. To follow these guidelines, the new UI would be distanced from a fixed position away from the user and would not follow the user when moving. The way it is set up is that when the user presses a button on the controller, the game menu will pop up some units in front of wherever the user is looking. A card row shows up with an image and a title (figure 7.5). A card represents a particular task, either a nail or a cutting task. The image and title showed depend on the task, making it easy for the user to get an overview of the different tasks and thus help with navigation. To interact with this game menu, the users are given a laser pointer on their right controller, which is only active so long as the menu is open. With this laser, the user can point to the option they want to explore and then press the trigger button on the controller to click on the UI element. The usability of this game menu is further improved by a hover effect which displays whenever the user points the laser at a UI element. For instance, if the user points at a task card, the background will change its color slightly to indicate that the user is currently pointing at this specific item and that the item is clickable.

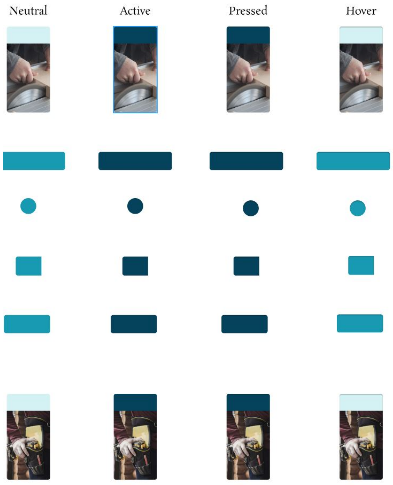


Figure 7.4: Components created in Figma

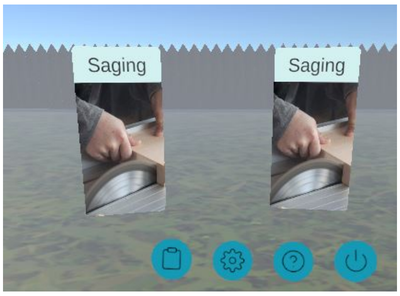
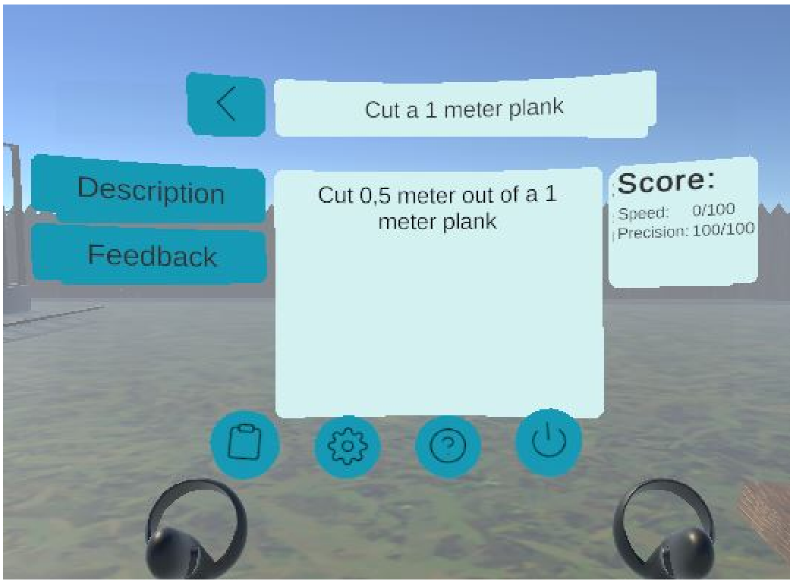


Figure 7.5: In game menu showing cards with tasks

Whenever the user clicks on a task card, they will be sent to a new page with all the information about the task (see figure 7.6). This includes the full title of the task, the description which describes what the user has to do to complete the task, and their current and max scores. In addition, there is a feedback and description button on the left. These can be clicked on to toggle between reading the task description and the feedback they have gotten for this particular task. Initially, the feedback message will be a standard one but will update as the user makes progress on the task. The score in the top right corner has two different scores: precision and speed. For the time being, only precision is measured.



Cut a 1 meter plank

Cut 0,5 meter out of a 1 meter plank

Score:
Speed: 0/100
Precision: 100/100

Figure 7.6: In game menu showing the description of a task. The menu looks slightly different in the Carpenter app when using a HMD.

To return to the task menu, there are two main ways of doing so. The first is to click on the back arrow in the top left corner of the task page. The second way is to click on the clipboard icon in the sub menu that hovers below the main menu (figure 7.6). This sub-menu also has other buttons; however, they do not have any functionality currently.

7.3.7 Tutorial

Due to a lot of end-users not having used VR before, and since it might not be apparent how the controls work for this app, a tutorial was made. This would be its own game scene and laid out similarly to the main game, although it would teach the player how to do different actions, such as picking up items and using the two existing tools. It would start by showing the player how to teleport. To illustrate this, an oculus controller would be shown in front of the player, and it would have the joystick animated, showing how to perform the teleport. After this, the player would be free to teleport to whatever station they wanted to learn about cutting or nailing. Due to time constraints, only the nail gun had a fully functioning tutorial. In this tutorial, there would be tooltips that explained to the user that the nail gun could be picked up with the trigger button on the controller and that the nail gun can shoot nails into planks. The nail gun has a custom animation where it nails two planks together, which is what was shown to the user in the tutorial.

Next to this animation were two planks and a nail gun so the player could test it out for themselves. As the tutorial was not completely finished, it was not utilized in the first user testing.

7.4 Results

The first user test was conducted in the IMTEL VR lab. In total, 11 users would test the carpenter app. The results of the first iteration are primarily the observations made and the users' feedback. The people testing were visitors to the IMTEL VR lab. The users consisted of employees from NAV and VR4VET partners. These are not the ideal users but helps highlight the main problems and strength of the app.

7.4.1 Observations

Since two students are working on this project, one would explain to the user what to do, while the other would write down the observations during testing. To ensure that the data collected could be used for this thesis, participants were asked to sign a consent document detailing what kind of information would be collected, how it would be stored, and what it would be used for. All of the 11 users agreed to sign this document and what follows are the observations that were made.

ID	Observation
O1	Users should not be able to move when they are using the task menu since this gives them 2 laser pointers when trying to teleport.
O2	The exit button on the game menu should have another symbol than a power off symbol. Perhaps replace it with a cross to indicate that the menu can be closed by pressing this button.
O3	Perhaps implement some sort of notification system when users have completed a task.
O4	Planks are really difficult to grab since the hitboxes are rather precise. Sometimes when cutting planks it is impossible to grab the smaller plank.
O5	Should not let users open the task menu when grabbing things since it just gets in the way and is most likely unintentional.
O6	The plank spawners are fidgety which makes it difficult sometimes to grab a new plank.
O7	Circle saw is too low on the ground. Probably happens since we were testing the app while sitting in a chair. Need to place the saw higher off the ground.
O8	The menu is opened a lot and probably by accident. Maybe require the user to hold down the menu button for a second or too before opening to prevent accidental menu opens.

O9	The plank does not follow the hand in a smooth matter, it kind of lags behind and is very jittery.
O10	Some users tries to use the teleport laser to grab planks for some reason. They also don't intuitively understand that when the laser is red it means that they can't teleport to that location.
O11	Some users thought the cutting block was the plank they were supposed to cut. The planks needs to be more visible or the cutting block needs to be remodelled so that it is not mistaken for a plank.
O12	Some users tried to grab the cutting block to move the plank towards the saw instead of using the wheel. Might have to replace the wheel with a handle on the cutting block to make it more intuitive to use.
O13	The plank spawner for the wall nail task should be made more noticeable. Users struggled with finding where to get planks. Some of them thought they had to use the plank props we had placed on the ground which were merely for aesthetics.
O14	Should place the nail gun closer to the wall task, since many people struggle with bringing it from the workbench to the house wall.
O15	It is possible to teleport on the house itself. This was problematic for some as they teleported way too far and had to turn around and try again.
O16	The nail gun can knock planks away which is bad, since this can ruin progress that the users have made on the wall nail task.
O17	Not all users completed the wall task as it was a bit difficult to do.
O18	The tool belt is sometimes in the way when using the nail gun. It snaps the nail gun when not intended by user. It is also not used by everyone as there are some users who just walks or teleports to the wall nail task while holding the nail gun in one of their hands.
O19	The teleporter cross inside the teleport marker sometimes gets stuck in objects like the plank spawner. Does not break any functionality but is a visual bug.
O20	Should not allow the user to place other things than planks in the circle saw. One user tried to place the nail gun in the cutting block and attempted to cut the nail gun with the circle saw.
O21	Maybe have an acceleration system for the walking movement so that users don't start with maximum velocity the instant they push the joystick forward. This would allow for more precise and fine movement when trying to position oneself for a task.

7.4.2 Feedback

After the users had tested the app, they were asked to give feedback if they had any. Despite not testing with many users, the feedback received varied regarding what could be improved or what they thought worked well. Some of the feedback contradicts each other such as one of the users claiming that it would be better to press the button once for grabbing, while another states that it's better to press and hold. These are the responses recorded from the users during and after testing the app:

- The plank was wobbly and difficult to handle.
- Make the tasks as easy as possible since the users who ends up using the product are those without education or work experience.
- I think the tasks should be divided into steps that the user can follow, since it is unclear what it is I'm supposed to do.
- The circle saw should be highlighted so that it is easier to see.
- There should be background noise from a construction site for a more immersive experience.
- Pressing once for grabbing would be better.
- Maybe have a video demo on how to perform the task.
- Why is the text in English?
- I got really dizzy from walking.
- The wheel on the circle saw is difficult to use. Once the cutting block has gone too far to the right, I struggle with getting it to go back to the left using the wheel.
- There should be some kind of success screen when completing a task.
- The circle saw is fine, however it could perhaps have more settings, for example have a setting that lets the user cut the plank at a 45-degree angle.
- The circle saw should be more detailed.
- Good idea to have a variety of plank types.
- The measure system should have a measuring tape to indicate which plank has been cut to the length displayed on the screen.
- Nailing should be visible. When I nailed planks I could not see where I have nailed. I can't see where I'm supposed to nail the last plank on the wall since I can't line up with the previous nails due to them being invisible.
- Tasks should be introduced with blueprints which highlights what you are supposed to make to make it more realistic.
- This app could integrate more professions into it, such as plumber and electrician since building a house is a job which requires several areas of expertise.
- The teleporter arc is opposite of what it should be. When pointing down the arc should come closer to the user and vice versa.
- Having an arrow at the end of the teleport arc makes no sense since the user can just turn after teleporting.
- It is more common to have to hold down the button for grabbing rather than

it being a toggle.

- There should be less space between the tasks, currently they are very far apart.
- The teleporting is not working while holding the nail gun.
- I find it impressive that it is possible to nail together planks.
- Walking with the left controller was so smooth, which I think is not natural / realistic.
- It would be more natural to grab items with the trigger button instead of the grip button, since picking up items in real life is usually not done without the index finger.
- Having hand models as controllers would be more natural
- It was fun to cut things with the saw
- Where am I supposed to place the plank that I have cut?
- There should be a way to inform the player where the materials are supposed to go after they have been cut.
- The circle saw should have some sort of security in place since it is a dangerous tool to handle in real life.
- I got a little dizzy from walking but did not get nauseous.

Chapter 8

Second Iteration

This chapter will explain the changes done in the second iteration as well as the results that were produced.

8.1 Updated Requirements

Here are some of the new requirements that were found after the first iteration had ended.

ID	Requirement
1	The UI needs to be reworked since it is often in the way or is not used
2	Tasks in the app should be simpler due to users struggling to perform them
3	The application should provide support for different languages and make it easy to both add new languages and switch between them

Table 8.1: The updated requirements for Carpenter VR.

8.2 Changes

8.2.1 Circle Saw

Although many of the users in the first user test were impressed by the circle saw, it became apparent during testing that it had numerous bugs which detracted from the user experience. Fixing it for the next iteration became, therefore, a priority. One of the major bugs in the previous version of the circle saw was that the snap zone attached to the cutting block would stop working on cut planks. When a cut plank is placed on the snap zone, what happens is that the rotation is set incorrectly, which leads to the planks being rotated 90 degrees when snapping. Another bug was that sometimes the snap zone would not update its position, making it impossible to move it and the plank it was holding. The second major issue with the circle saw was that the smaller cut planks could not be picked up. This made removing the smaller plank impossible after cutting it in two. Finally, the texture of the cut planks would sometimes glitch out, presumably because of how the cutting script was implemented. All of these things led to a complete redesign of the circle saw in the way it functions.

Much of the feedback in the previous iteration stated/indicated that the tasks in the app should be simpler for them to become more usable by end-users. With this in mind, a new way to cut planks was implemented. Instead of having a script that can cut any shape, a new script was made that only works for planks. This script is less versatile as it can only cut planks, but since it is more simplistic, it is less prone to unintended behavior. This new script will check if the saw has come into contact with a plank by checking if the collided object has the sliceable script attached. If this is the case, it will cut the plank in 2 whenever the saw hitbox leaves the plank hitbox. The original plank is destroyed to sell the illusion of cutting the plank into smaller planks, and two new planks are instantiated in the previous plank's position. These new planks will have lengths different from the original plank, which are calculated just as the cut happens. The sliceable script was modified so that it is possible to set the length of the plank to the desired value. This is what's used after instantiating the two new planks to have the appropriate length. A final touch is that the planks are separated by a small distance, thus making it easier to see that the plank has been cut. An improved wheel was 3D modeled to make it easier to understand that the "wheel" was able to rotate. This affordance was not obvious; thus, the "wheel" needed a signifier. This reduced the number of users not understanding how to change the distance when cutting the planks.

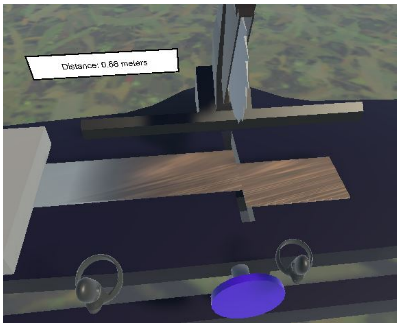


Figure 8.1: Circle saw cut plank from first iteration.

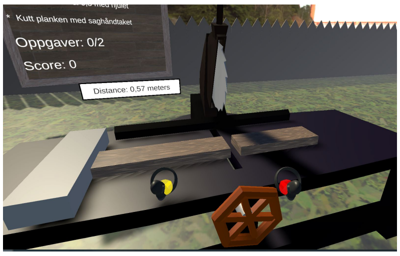


Figure 8.2: Circle saw cut plank from second iteration. Improved "wheel" in the bottom center.

8.2.2 Nail Wall Task

The task of nailing planks to the wall had numerous flaws, mainly when the users had to place the plank on the wall itself. Three things mainly hindered this. The first was that they had to grab a plank and teleport it to the wall to place it. This was an issue since the house they were nailing the plank to was teleportable, which means that some users ended up teleporting inside the house and sometimes even further. The second problem was placing the plank on the wall. To do this, the user had to put the new plank on top of the previous plank on the wall. Getting the new plank to stay where it was supposed to be, was problematic for some, which led to the plank falling off the wall. The third and final issue was the nail gun and how it interacted with the plank. Since all grabbable objects have a rigidbody component, it is possible for them to collide and interact with one another. This was an issue because when the users tried to nail the plank to the wall, it was probable that the nail gun they were holding would collide with the plank they were trying to nail. This collision would apply a force on the plank, thus accelerating it and making it fall to the ground.

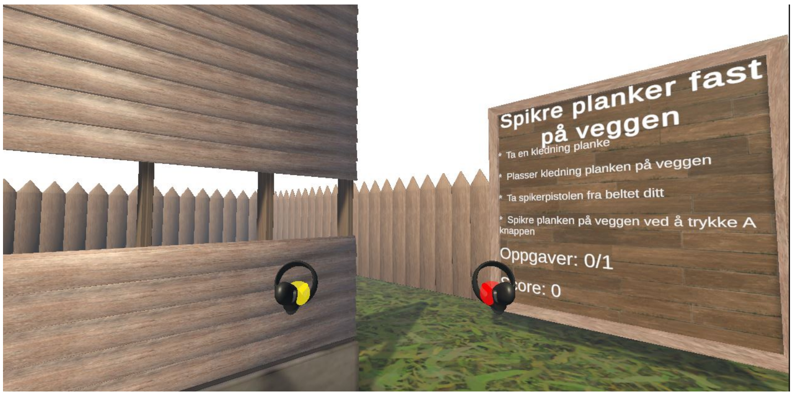

Spikre planker fast på veggen
* Ta en kledning planke
* Plasser kledning planken på veggen
* Ta spikerpistolen fra beltet ditt
* Spikre planken på veggen ved å trykke A knappen
Oppgaver: 0/1
Score: 0

Figure 8.3: Image of the nail wall task and the board describing the task

To fix these issues, constraints were added to the task. Movement, in general, was more restrictive (see 8.2.6), which removed the possibility to overshoot when trying to teleport in front of the wall. The nail gun could no longer affect the planks in any significant way since now the planks had their mass increased so drastically that it became impossible to hit them over with the nail gun. Lastly, the slots where the planks are placed now have snap zones which will forcefully place the plank in the exact spot it needs to be. This leniency makes it very easy to place the planks, and it is no longer possible to hit them over with a foreign object. The only way to move planks after they have snapped into place is to grab them.

8.2.3 Toolbelt

The toolbelt has seen a slight but essential change since the previous iteration. Before, it started empty, and the user could decide whether or not they wanted to put tools in it. It was noticed during testing that not everyone opted to use this functionality and instead just grabbed the nail gun and teleported while holding it. Among this, there was a problem where many users frequently dropped the nail gun on the ground by accident, forcing them to crouch down to pick it up. To remedy this, the nail gun would automatically slide back into the toolbelt when it was not in use. This made it impossible to lose the nail gun since dropping it would make it fall back into the toolbelt. This also increased the convenience since the nail gun was always at hand and easily reachable wherever the player went.

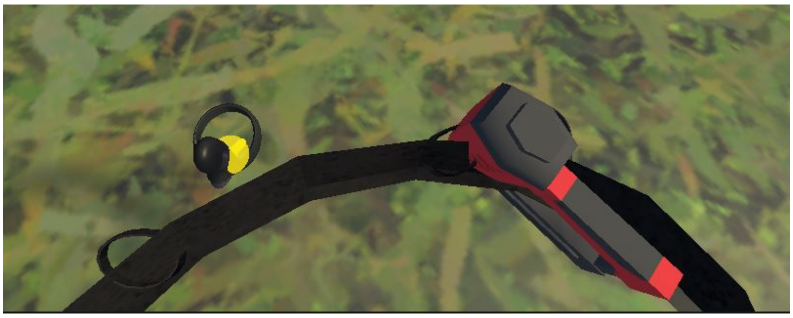
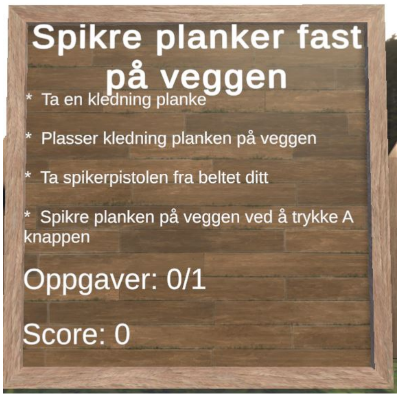


Figure 8.4: Image of the toolbelt from the players perspective. The nailgun is in the toolbelt

8.2.4 Task Menu

As previously mentioned in 7.3.6, a game menu was added. In this menu, there was a list of different tasks the users could navigate through to get an explanation and feedback on how they had fared. The previous testing made it clear that this menu was not used as hoped and intended. Many users forgot about it, and since it can only be accessed by pressing a certain button on the controller, it did not feel like it was very well integrated into the app. Therefore, the task menu was reworked to be accessible in the game without consciously deciding to open it. To do this, a *task board* was created. These were physical boards in the game with text that would update as the player progressed with tasks. Each task would have its own board containing information about the corresponding task. These worked very much like the previous task menu; however, these would appear in the world as soon as the player teleported to a workstation. For instance, when the player would teleport to the circle saw task, the board containing its information would fly up through the ground and appear in the user's view. This board holds information about the tasks, including their title, description, and the number of points the user has accumulated. One of the users in our previous user test pointed out that it would be helpful if there were instructions on what one needs to do to complete the task. This was implemented using this board system by writing the task description as a series of steps.



Spikre planker fast på veggen

* Ta en kledning planke

* Plasser kledning planken på veggen

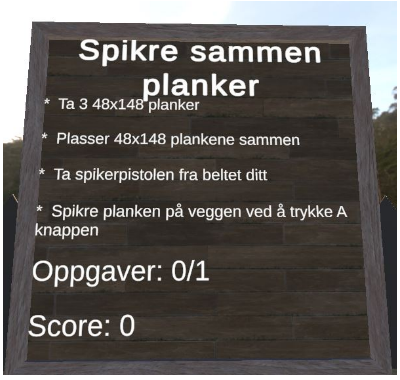
* Ta spikerpistolen fra beltet ditt

* Spikre planken på veggen ved å trykke A knappen

Oppgaver: 0/1

Score: 0

Figure 8.5: The board describing the wall task



Spikre sammen planker

* Ta 3 48x148 planker

* Plasser 48x148 plankene sammen

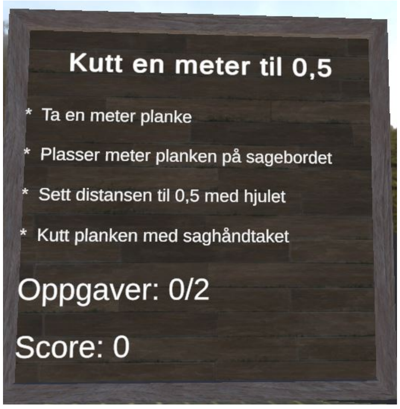
* Ta spikerpistolen fra beltet ditt

* Spikre planken på veggen ved å trykke A knappen

Oppgaver: 0/1

Score: 0

Figure 8.6: The board describing the workbench task



Kutt en meter til 0,5

* Ta en meter planke

* Plasser meter planken på sagebordet

* Sett distansen til 0,5 med hjulet

* Kutt planken med saghåndtaket

Oppgaver: 0/2

Score: 0

Figure 8.7: The board describing the sawing task

8.2.5 Language Manager

One of the common questions from the users' testing was why the application was in English. The reason for this at the time was that it was expected that English-speaking users would test that day, but the question still highlights an important aspect of the game – multiple language support. This was implemented by having a language system where all texts are now defined in a custom class called *Sentence*. This sentence class has multiple strings, each corresponding to its respective language. This way, when adding text, it can be written in various languages by creating a sentence object. Currently, it only supports Norwegian and English, although it should be possible to tweak it to support more languages. All existing text in the app was then replaced with sentence objects containing a Norwegian and English versions. For the app to know which version to use, there is a global variable that tells the app what the current language is. The Sentence class uses this public variable when deciding what string to use for the text in the game.

8.2.6 Movement

In the theme of making everything simpler, the movement was reworked slightly. Before, either walking by using the joystick on the left controller or teleporting using the joystick on the right controller. In this new version, it is only possible to teleport. The main reason for this is that the walking system was too smooth, making many users sick when using it. This was a case of it being completely fine, or the user would find it nauseating and thus would not use it. One of the people who tested the app in the previous iteration was a game designer for VR. One of their claims was that sliding movement like the one implemented does not work for VR games, and that teleportation should be the standard. Due to this feedback, it was decided that walking should just be removed and replaced with teleporting. Therefore, it is now only possible to teleport, and it is something that can be done with either joystick.

Furthermore, the teleport has become more restrictive as what is now considered *teleportable* has changed. Before this iteration, users were allowed to teleport on everything marked as the ground. This covered a large area, and the users could wander far away from the task stations if they desired. The problem was that many users would struggle to find a good spot to stand when performing the tasks. This led to them being located in suboptimal positions for completing the task. Our solution for this problem was to have fixed spots they could teleport to. These were indicated by having small blue circles in the ground showing the users where they could teleport. These teleport spots would have snapping, meaning that when the laser from the teleport came close enough, it would snap the laser to the teleport spot, making it easy to maneuver between workstations. A final tweak to the teleport was to activate it by pressing down on the joystick. This means that it is no longer sufficient to push the stick in the direction you want to teleport, but you also have to press the joystick. Once a location has been chosen, the user has to stop pushing the joystick for the teleport to activate. This change made the controls more like other Immersive Job Taste applications.

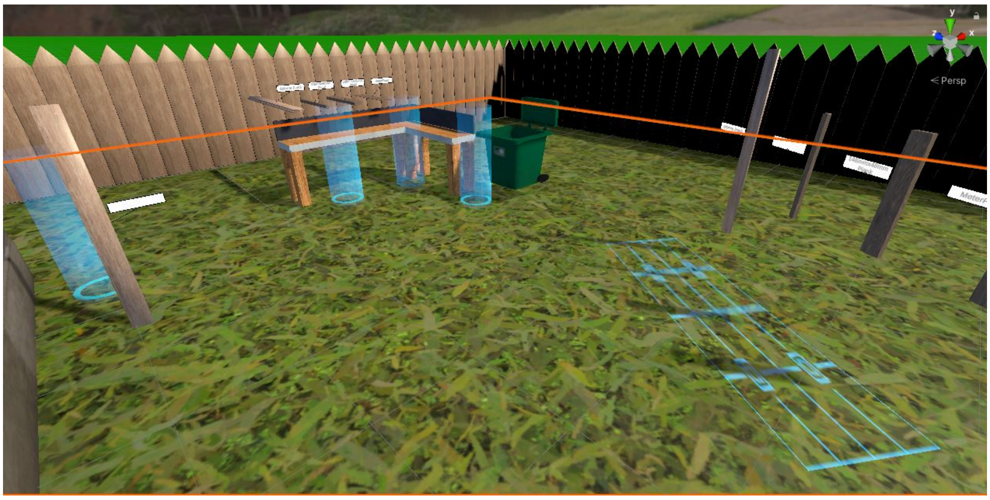


Figure 8.8: Image showing where the player can teleport. The blue cylinders and blue pattern near the planks can the player teleport to.

8.3 Results

Three user tests were conducted. During these days, 4, 3, and 11 users were testing the app. Nine of the users also answered a questionnaire 8.3.3. The people testing were visitors to the IMTEL VR lab. The users consisted of teachers wanting to understand better how VR can be used in their field. These are not our ideal users as they would be young job seekers, but they are helpful in testing and improving our app.

8.3.1 Observations

ID	**Observation**
O1	The new task board is not visible enough which lead to many users not seeing it before we pointed it out.
O2	Teleport was a bit difficult to understand for some, although others managed it well.
O3	The workbench task board contained incorrect instructions, stating in one of the steps that the user had to nail the plank on the wall.
O4	There was a bug with the snap zone on the wall task where 2 snap zones would overlap since they were too close to each other. This made it impossible to place planks on the wall, since this overlap caused the top plank to fall off every time.
O5	The controls of the game needs to be explained better.

O6	The instructions on the board are working as intended. Some of the users are able to complete all the tasks with no or minimal guidance from us.
O7	Planks are an issue to grab and manipulate on the workbench tasks as the spawned planks tend to collide and get stuck in the top plate of the workbench. Often this led to planks being lost on the ground.
O8	The planks are a bit glitchy. When they are stacked on top of each other they tend to merge into one another due to them having such a large mass.
O9	The cutting block is once again mistaken for a plank as users are trying to cut it.
O10	Many users had to ask where the planks are on the saw task. These should be made more visible, so people know immediately where to go to get planks for cutting.
O11	Some users used quite a bit of time to figure out where the wheel was on the circle saw. It might be a good idea to relocate it to somewhere else.
O12	The cutting block seemed to make more sense once they had attached a plank to it.
O13	The circle saw is very inconsistent at cutting planks. Sometimes it cuts them as soon as the blade touches it, while other times it takes more effort. Related to this are the sounds produced by the circle saw. Since it is based on collision with the plank it sometimes loops sounds frequently because of frequent collisions.
O14	Planks are generally a bit hard to grab due to users not understanding that they have to reach their hand / controller inside the plank before they are able to grab it. It might be a good idea to have some sort of distance grab to make it easier to grab things.
O15	Some users are moving the cutting block without attaching a plank to it. This again indicates that it is not clear what the cutting block actually is, and many are mistaking it for a plank.
O16	Some users are wondering why there is trashcan next to the workbench. Currently it has no purpose so it should be removed.
O17	All of the users needed guidance for knowing which button is used to fire the nail gun. Some sort of indication should be made so that it is clear which button to use.
O18	One of the highlights on the snap zone on the wall got stuck and was glowing indefinitely.

O19	Some users would press in both of the joysticks at the same time when teleporting. This led to a bug where the users would manage to teleport outside of the designated playable area, thus leading to a state in which the player could not teleport back which led to us having to reset the game.
O20	One of the users tried to walk over to the circle saw with the plank instead of teleporting.
O21	The nail gun is often in the way and is grabbed by accident when the user is trying to grab something else.
O22	Some of the users did not fully read the task on the board and we had to remind them that the plank needs to be cut to a certain length. Sometimes the same was true for plank types.
O23	The spacing between the title and description on the task boards are too small. These should be spaced out a bit better.
O24	There are very few users that understood that when the green highlight zone appeared, it means that they can drop the plank and it will snap into place.
O25	There is a small subset of users who tried to grab by simply pressing the grab button once instead of holding it down.
O26	The nails are not visible when nailing the planks on the workbench task, only for the wall nailing task.
O27	Some of the users would not finish the app and would only complete 1 or 2 tasks.
O28	It is not entirely self-evident that the planks are supposed to be placed in the cutting block.
O29	One of the users managed to nail one of the planks into the circle saw by accident. This happened due to grabbing the nail gun by accident and pressing the fire button.
O30	The distance tooltip is missing a Norwegian translation.
O31	The tooltips are very inconsistent in displaying their text. It only shows from certain angles / distances.
O32	There seems to be a bug where the nail gun is dropped unintentionally, making it difficult to use.
O33	One of the users thought they were supposed to place the plank to the left of the wall instead of the gap in the wall.
O34	A lot of the tasks became hard or easy depending on the height of the user. The circle saw in particular was very difficult to use for tall users which led to them having to bend their knees a bit to use it.
O35	There is a bug where the wall task can be completed without having to nail all 3 planks.
O36	The old task menu is often opened by accident since the button is right next to the button that fires the nail gun.

O37	A lot of users would lose their plank when teleporting to the circle saw.
O38	Planks on the workbench teleport back on the table when falling on the ground, but nailed planks do not.

8.3.2 Feedback

As in the previous iteration, users were asked to give feedback regarding the app.

- The saw is very high up, an unrealistically high circle saw.
- I think it was understandable when it comes to what it is I was supposed to do.
- There should be some sort of warning whenever you are doing something dangerous.
- The circle saw should have some sort of protective shield to make it more realistic.
- When I was nailing the planks together I was wondering how long the nail would be.
- It took some time to get used to the controls and game mechanics, but when I arrived at task 2 and 3 I had understood how the app was built with regards to teleporting and such.
- I thought it was difficult to execute actions since I am not used to using VR headset and controllers.
- It was clear what I was supposed to do on the tasks. Having instructions really helped.
- I did not see any nails when using the nail gun.
- The nail gun was sometimes in the way when doing tasks.
- I found the tasks understandable, and I thought the graphics were quite nice.
- It was a bit illogical to nail together 3 planks on top of each other.
- It would be helpful if there were instructions on where the plank should be placed on the wall, if it should be at the bottom or the top.
- I got a bit dizzy from using the app.
- I thought the app was good, but there was no explanation on how to shoot with the nail gun.
- I found it very disturbing with the fences surrounding the map. I felt like I was boxed in.

8.3.3 Questionnaires

At the end of the test, users were asked if they were willing to fill out a questionnaire. Due to not having an English version of the questionnaire, users on

April 26th could not answer the questionnaire. Furthermore, not everyone who participated sent in their answers leading to less feedback than ideal. In total 9 answers were received from the questionnaire. The questionnaire focuses on the user experience and uses the Likert rating scale.

ID	**Question**	**Average Score**	**Median Score**
Q1	I liked using the carpenter app with VR headset	4.3	4
Q2	It was easy to understand what I was supposed to do in the carpenter app	4.4	4
Q3	It was easy to understand the different work tasks in the carpenter app	4.6	5
Q4	Things (tools, items, etc) in the carpenter app behaved in an expected manner	3.7	4
Q5	It was easy to read the text that appeared in the app	4.2	4
Q6	The carpenter app gave me more insight into what the profession is about	3.7	4
Q7	I'm thinking that I want to use the carpenter app regularly	3.3	3
Q8	I found the carpenter app unnecessarily complex	1.3	1
Q9	I thought the carpenter app was easy to use	4.0	4
Q10	I think that I would need the support of a technical person to be able to use the carpenter app	2.2	2
Q11	I found the various functions in the carpenter app were well integrated	3.9	4
Q12	I thought there was too much inconsistency in the carpenter app	2.3	2
Q13	I would imagine that most people would learn to use the carpenter app very quickly	4.2	4
Q14	I found the carpenter app very cumbersome to use	1.7	2
Q15	I felt very confident using the carpenter app	4.1	4
Q16	I needed to learn a lot of things before I could get going with the carpenter app	1.9	1
Q17	I think that the job taste apps is a good tool for career guidance	4.0	4
Q18	Such job taste apps should be a part of NAV's offers for job seekers	4.1	4

Q19	NAV needs more job taste apps for more professions	4.0	4

SUS Score

ID	**SUS score**	**Grade**	**Adjective**	**Acceptable**
1	80.0	A-	Good	Acceptable
2	65.0	C	OK	Marginal
3	80.0	A-	Good	Acceptable
4	62.5	D	OK	Marginal
5	75.0	B	Good	Acceptable
6	77.5	B+	Good	Acceptable
7	92.5	A+	Best Imaginable	Acceptable
8	80.0	A-	Good	Acceptable
9	65.0	C	OK	Marginal

This yields an average SUS score of:

$$\overline{SUS} = \frac{80.0 + 65.0 + 80.0 + 62.5 + 75.0 + 77.5 + 92.5 + 80.0 + 65.0}{9} = \frac{677.5}{9} \approx 75.3$$

Chapter 9

Third Iteration

This chapter will explain the changes done after the second iteration and present the results of user tests conducted on the third iteration of the carpenter app.

9.1 Changes

Here are the major changes that were made in this iteration. Some of the changes are minor bug fixes. Most of the bugs found in the user tests from the second iteration were fixed, although some might still exist. Not all glitches were possible to reproduce, but most of them should be fixed or ironed out. Most of the changes done, however, were the redesign of things that did not work well enough in the previous iteration, such as the workbench task and the tutorial.

9.1.1 Redesigning the Workbench Task

One of the things observed during user testing in the second iteration was that the workbench task did not seem to make very much sense. This was also reflected in some of the feedback: "It was a bit illogical to nail together three planks on top of each other." This led to redesigning the workbench task. Instead of having a task where the user has to nail three planks together as they see fit, a task was made where they had to create something specific using the nail gun. This led to a new task where the user creates a frame and places it near the house in the scene that is being constructed. This was a more realistic and organized work task that utilized most of the tools created. This was implemented by having three snap zones for each plank in the frame. Each snap zone has a green highlight zone for the indication (see figure 9.1), and initially, only one would be active. The first snap zone would require the plank to be cut to 0.5 meters while the remaining 2 – the legs, would be 1 meter. When the user cuts a 48x148 plank to 0.5 meters and places it in the first snap zone, it will trigger the other two snap zones, making them visible and enabled. When the user places a one-meter 48x148 plank in each snap zone, they have to nail the three planks together to create the frame. Finally, a snap zone with the appearance of a frame would be placed by the house in the scene to indicate where the user has to put the newly created frame. The task would be completed after placing the frame in the snap zone by the house. A new task manager, the frame task manager, was created to keep track of the task, making it possible to tell the user that they had completed the task.

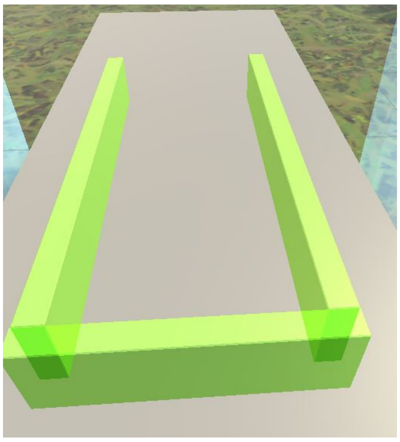


Figure 9.1: Green highlight zone for all three planks.

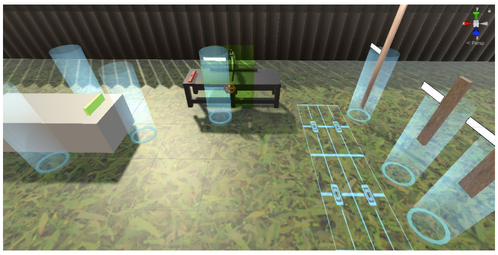


Figure 9.2: The new layout for the frame task

9.1.2 New Tutorial

Although a tutorial had been implemented, it became apparent that it would not suffice based on the results of the user tests. In retrospect, it was very similar to the main app in that it is very open, making it possible for users to wander wherever they want. This leads to a poor user experience mainly because it would be a lousy tutorial if it explained to the user how to cut planks first if the user does not yet know how to grab things as it is required to grab planks and put them on the circle saw before they can cut it. Due to this reason, the tutorial was reworked. In the new tutorial, the user is placed in a closed room connected to other rooms. This creates a progressive tutorial where the most basic concepts are explained first, leaving the more difficult ones at the end. There are four rooms in this tutorial, and each room presents a specific mechanic of the app. Initially, the user is locked inside the first room, and to proceed to the next room, the user has to complete a task given by that room. Once the user completes a task given by a room, the next room will be accessible by way of a door that gets unlocked. The door is blocking the way to the next room and will only open once the user completes the task. The 4 rooms (in order from start to finish) explains the following mechanics: Teleportation, grabbing, nailing, and cutting.

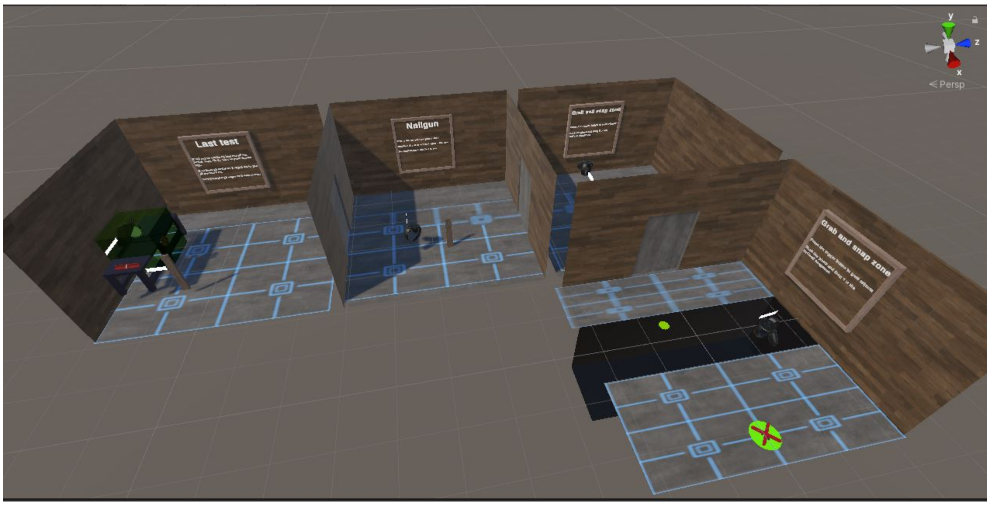


Figure 9.3: All four tutorial rooms seen from above.

The teleportation tutorial is straightforward. The user is placed in the back of a room and is given instructions on how to teleport. All explanations are done in text form using the boards created in the second iteration (see chapter 8.2.4). In the teleport tutorial, they have to cross a gap in the floor by teleporting over to the other side. In addition to the board explaining what to do, there is also a 3D model of an Oculus controller alongside a tooltip, which illustrates to the user how it is done on their controller. Together with the 3D model is an animation showing that the joystick on the controller is being pressed in to make it more obvious what they have to do for the user. When the user teleports over the gap in the floor, the door to the next room opens.

The grabbing tutorial teaches the user how to pick up items and place them. In this room, a plank needs to be moved to a designated snap zone highlighted by a green outline. All instructions are given by the board, as well as another 3D model of an Oculus controller, which has an animation where the grab button is being pressed. Once the user has grabbed the plank and placed it in the snap zone, they can proceed to the nailing room.

For the nailing tutorial, the user has to nail a plank that has been placed in the room. Explanations on how to do this are given alongside a model of the controller with text stating which button must be pressed to fire the nail gun. After nailing the plank at least once, the user can proceed to the cutting tutorial.

The final room in the tutorial explains how to use the circle saw to cut planks. Here the user has to use several things they have learned so far to complete the task. The task given is to cut a plank to a certain length. The plank can be grabbed from a plank spawner in the room. Then the user must take the plank, move it over to the circle saw, and place it in the appropriate spot. They then have to adjust the plank length by turning the wheel on the saw. Finally, they cut it by grabbing the handle of the saw. All of these steps are explained by the board in the room. When the plank has been cut, the tutorial is over. The board in the cutting room will have instructions for the user, explaining that they need to point the laser on the board to start the main application.

9.1.3 Improving the circle Saw

The circle saw worked mostly as intended during testing. However, some minor things needed to be fixed to make it more consistent with use. First off was the cutting block. As noted in the observations, this block was often mistaken for a plank, leading to some users trying to cut it. A proper 3D model was created to remedy this, which looks more like a place where the user can place planks. It is much thinner than the previous cutting block and has a bright red color, making it unlikely to be mistaken for a plank (figure 9.4). Another issue with the circle saw was the inconsistency when it comes to cutting. Some users would cut the plank instantly as soon as the saw touched the plank, while others had act use some effort to cut the plank. The logic for cutting planks dictates that a plank will only be cut when the saw hitbox leaves the plank hitbox. This works fine if the user cuts straight down. However, if the user cuts into the plank and then immediately raises the saw, the condition is true, and the plank is cut instantly. This is not realistic and can be a bit underwhelming when using the saw. This problem was fixed by requiring the user to cut a certain distance down before the plank is split in 2. In other words, the circle saw has to rotate a predefined amount before it allows the plank to be cut when the saw hitbox leaves the plank. Lastly, a final touch was added to the circle saw to make it more realistic. A dust particle system was imported from the Unity Asset Store. This particle system was added to the circle saw and would activate whenever a plank was cut. This means that whenever a plank is being cut, it starts to spew out dust from it, simulating the sawdust that would appear in real life.

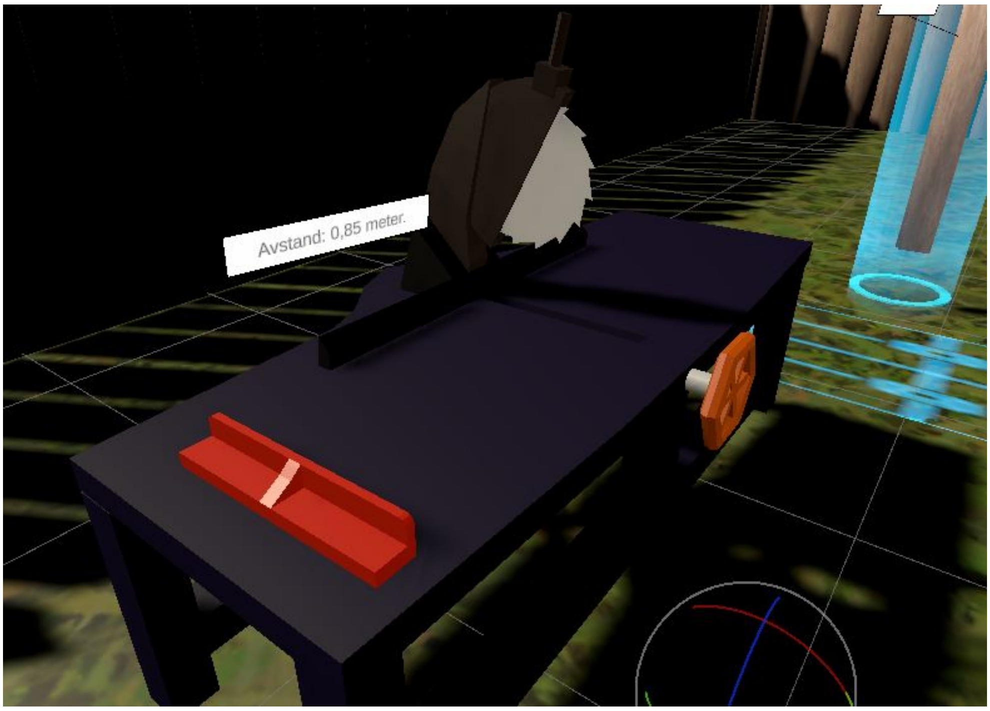


Figure 9.4: Circle saw with improved cutting block. Cutting block is the red object to the left

9.1.4 Miscellaneous

This section will highlight some minor but important changes that were done.

Task Manager Refactorization

The task manager system was slightly optimized and reworked. Previously, whenever a script needed access to the task managers, they would need a reference to it or would have to search the entire scene to find the game object holding the task manager they were looking for. Both of these methods had major flaws. The former was not ideal as adding references to the task managers would introduce changes to the game scene, which can lead to merge conflicts. Secondly, adding references for every object that needs to access a task manager is not convenient. The latter is not very performant, especially when it is done often. As the app is not lightweight, it must run as smoothly as possible. To fix both of these issues, a singleton solution was used. This made it so that the task managers would be grouped up in a single object which is static, meaning that it can be referenced from anywhere without having to add a reference. This made it much easier to work with the task managers and call on them when creating and validating tasks.

Fixing Tooltips

There was a bug where tooltips would only be visible from certain angles. This was especially a problem for tall users since they would be at an angle where the distance text on the circle saw would not be visible, making it impossible for them to know the distance. The same problem occurred for other objects with tooltips, such as the plank spawners. Sometimes the text on the plank spawners would disappear depending on the angle it is looked at. A fix for this problem was to make the tooltips follow the VR headset in the game world. The tooltips have a script with functionality for tracking a game object. When adding the VR headset game object as a reference in this script, the tooltips will rotate with the player so that it's always facing the player.

Removing 48x48 plank

Due to a lot of confusion with the plank types, the 48x48 plank was removed entirely. One of the reasons for this was that this plank was not used for any particular task. However, the main reason was that this plank was often confused with the 48x148 plank. These two plank types have very similar names, so it was decided that it would be better to remove one of them.

9.2 Results

Results are from two user tests. One of the user tests is our primary user group, and the other is NAV employees. Both tests resulted in 10 combined answers to the questionnaires, with five from each of the tests. The SUS score shows that both the user groups give the app a high usability score.

9.2.1 Observations

ID	Observation
O1	Some users tried to nail the oculus controller instead of the plank in the tutorial.
O2	One user in particular adjusted the distance of the cutting block before placing the plank.
O3	One of the users that had tried VR before managed to complete the app without any guidance and significantly faster than any previous tester.
O4	Users tend to forget pretty quickly what they were supposed to do and has to look at the board a lot. This was a problem for the frame task since the board disappears when teleporting to the circle saw. The board should just be active at all times to mitigate this.

O5	The frame task board did not have a complete task description. There was no step indicating that the user has to nail together the plank, leaving the user confused.
O6	It was difficult to see the boards in the tutorial.
O7	There was a glitch where distance grab was enabled which lead to unintended behavior such as users picking up planks from a distance when they intended to pick up the nail gun.
O8	One of the users gave up after the wall task and did therefore not try the frame task.
O9	It should be clearer when one has managed to complete a task in the tutorial. One of the users were confused as to if the task had been completed or not.
O10	Some users struggled immensely to leave the tutorial by pointing the laser at the board. For some reason the hitbox was extremely difficult to hit.
O11	It might be a good idea to tell the users where on the wall they need to nail the plank as some of them are asking where they should nail.
O12	The exterior plank spawner by the wall task is in the way when trying to teleport to the workbench.
O13	The frame task is a bit complex and should probably be split into several steps that when completed gives feedback to the user so that the user knows that they are on the right track.
O14	A lot of users tried to grab the Oculus controller 3D model in the tutorial for grabbing instead of grabbing the plank.
O15	It can sometimes be difficult to place the plank on the circle saw in the tutorial due to it getting stuck in the saw because they don't lift the plank high enough.
O16	Planks are frequently dropped by accident when teleporting with it.
O17	The snap zone hitbox for the circle saw glitched completely out for one of the users. This led to the plank being snapped incorrectly. Although the reason is unknown, it might have something to do with distance grab being enabled as that caused a lot of problems for several tasks.
O18	Despite removing the 48x48 plank, there are still a handful of users who grabs the wrong plank. There also seems to be no rhyme or reason to this as some users pick up the exterior plank while other pick up the meter plank.
O19	A few users would try to place the plank at the middle of the cutting table instead of placing it by the cutting block.
O20	We forgot to fix one of the tooltips to follow the user, namely the circle saw in the tutorial. Thus, the distance text would be invisible for some of the users.

O21	There is an unintended mechanic on the frame task where users can grab the first plank they placed after one or both of the other planks have been set. This should not be allowed as it made some of the users lose progress since they grabbed the wrong plank and undid some work that they had done.
O22	One of the users struggled a lot with the teleport concept. Having to press and hold, then aim is maybe not intuitive enough.
O23	The 3D models of the Oculus controllers in the tutorial gets a lot of attention. This is sometimes a problem since it steals a lot of attention away from the board which is what actually explains to the user what they need to do.
O24	Some of the users were confused on how to get to the next room.
O25	Not everyone reads the text on the boards and proceeds to wing the task.
O26	One of the users managed to grab the wrong plank every single time on the frame task (3 in total).
O27	For some reason the nail gun is able to knock snapped planks away on the frame task.
O28	One of the users gave up after the frame task and did not attempt the wall task.
O29	Not all users are nailing the frame in the intended spot which leads to some very ugly frames being made. An indication should be added which shows the user where they should nail.
O30	Testers are confused where the tasks are in the main app. There should be some waypoints or other indicators to convey this.
O31	It should be easier to move from the frame task to the wall task. Currently the place where the user puts the frame is very isolated and to get to the wall task they first have to teleport to the workbench.
O32	There are some grammar mistakes on the board on the wall task.
O33	On the English version of the app, the tooltips have text sticking out from their boxes.
O34	The planks should be moved more forward on the grab tutorial to make them more visible as they are not easy to spot. The same should be done for the nailing tutorial.
O35	There is a bug where it is somehow possible to complete the cutting tutorial despite not cutting the plank. It was completed, but all the user had done was placing the plank on the cutting table.
O36	One of the users tried to grab the green highlight zone on the frame task.
O37	The teleport points on the plank spawners are placed badly as they are difficult to hit with the laser pointer.

O38	The text on the frame task board should probably specify that the nail gun is located in the user's belt as one of the users was wondering where the nail gun was.
O39	Some users are still trying to move the cutting block by grabbing it instead of using the wheel.
O40	A couple of users tried to start from the top instead of the bottom on the wall task. Might be a good idea to tell them that they have to start from the bottom.

9.2.2 Feedback

Here are some of the feedback received in no particular order:

- The app was fun to use
- This version was much better than the first version I tried (this was a user that had tried the app in an earlier stage)
- There was a bit of confusion in the tutorial. I was not completely sure where I was supposed to go.
- It was a good tutorial on explaining how the app works.
- It was a bit difficult to get used to using the app.
- Challenging to figure out where you're supposed to go. If there were arrows which showed the way it would be much easier.
- It was much better to have these task boards instead of the tablet that the other job taste apps are using.
- I haven't done this before, so I didn't know where to press on the controller and this took up a lot of my attention.
- It was a bit cool to turn the wheel when adjusting the distance of the plank.
- The toolbelt was too big in my opinion, it should be narrower.
- It's good that there are green highlight zones to show where things are supposed to be placed.
- It was a bit unusual for me to teleport in order to move in the VR world.
- I think the app was very realistic.
- I had to get used to the distances in the game world.
- The app was intuitive.
- It was a bit confusing to know what I was supposed to do, but once you explained to me that I have to read the boards it became a lot easier.
- I have not done this before, but I managed to quickly get into it.
- It was nice with the instructions telling me what I was supposed to do and such.
- In the beginning I forgot what buttons I was supposed to press, but it became easier to remember after using the app for a while.
- It was pretty realistic, it felt as if the things I did were real.

9.2.3 End Users Questionnaire

Similarly to the previous iteration, users were asked if they would be willing to fill out a questionnaire B.1.

General Questions

ID	Question	Average Score	Median Score
Q1	I liked using the carpenter app with VR headset	4.2	4
Q2	It was easy to understand what I was supposed to do in the carpenter app	4.2	4
Q3	It was easy to understand the different work tasks in the carpenter app	4.2	4
Q4	The tutorial in the beginning gave me enough knowledge to use the carpenter app	4.2	4
Q5	Things (tools, items, etc) in the carpenter app behaved in an expected manner	4.2	4
Q6	It was easy to read the text that appeared in the app	3.6	4
Q7	The carpenter app gave me more insight into what the profession is about	4.0	4

System Usability

ID	Question	Average Score	Median Score
Q8	I'm thinking that I want to use the jobbtaste apps regularly.	4.2	4
Q9	I found the jobtaste apps unnecessarily complex	2.0	1
Q10	I thought the carpenter app was easy to use	4.8	5
Q11	I think that I would need the support of a technical person to be able to use the carpenter app.	2.4	2
Q12	I found the various functions in the carpenter app were well integrated.	4.6	5
Q13	I thought there was too much inconsistency in the carpenter app	2.2	2

Q14	I would imagine that most people would learn to use the carpenter app very quickly.	4.4	4
Q15	I found the carpenter app very cumbersome to use.	1.8	1
Q16	I felt very confident using the carpenter app.	4.4	5
Q17	I needed to learn a lot of things before I could get going with the carpenter app.	1.4	1

SUS score end users

ID	SUS score	Grade	Adjective	Acceptable
1	95.0	A+	Best Imaginable	Acceptable
2	65.0	C	OK	Marginal
3	87.5	A+	Best Imaginable	Acceptable
4	92.5	A+	Best Imaginable	Acceptable
5	67.5	C	OK	Marginal

Table 9.4: SUS score for each user alongside the corresponding grade, adjective and whether or not the result is acceptable

This gives an average SUS score of:

$$\overline{SUS} = \frac{95.0 + 65.0 + 87.5 + 92.5 + 67.5}{5} = \frac{407.5}{5} = 81.5$$

9.2.4 NAV Employees Questionnaire

Similarly to the previous iteration, users were asked if they would be willing to fill out a questionnaire B.2.

General Questions

ID	Question	Average Score	Median Score
Q1	I liked using the carpenter app with VR headset	4.8	5
Q2	It was easy to understand what I was supposed to do in the carpenter app	4.6	5

Q3	It was easy to understand the different work tasks in the carpenter app	4.6	5
Q5	Things (tools, items, etc) in the carpenter app behaved in an expected manner	4.6	5
Q6	It was easy to read the text that appeared in the app	4.8	5
Q7	The carpenter app gave me more insight into what the profession is about	4.2	4

System Usability

ID	**Question**	**Average Score**	**Median Score**
Q8	I'm thinking that I want to use the jobbtaste apps regularly.	4.4	4
Q9	I found the jobtaste apps unnecessarily complex	1.2	1
Q10	I thought the carpenter app was easy to use	4.6	5
Q11	I think that I would need the support of a technical person to be able to use the carpenter app.	1.8	1
Q12	I found the various functions in the carpenter app were well integrated.	4.4	4
Q13	I thought there was too much inconsistency in the carpenter app	1.4	1
Q14	I would imagine that most people would learn to use the carpenter app very quickly.	4.4	4
Q15	I found the carpenter app very cumbersome to use.	1.2	1
Q16	I felt very confident using the carpenter app.	3.8	4
Q17	I needed to learn a lot of things before I could get going with the carpenter app.	1.6	2

SUS score NAV

ID	SUS score	Grade	Adjective	Acceptable
1	70.0	C	OK	Marginal
2	87.5	A+	Best Imaginable	Acceptable
3	95.0	A+	Best Imaginable	Acceptable
4	80.0	A-	Good	Acceptable
5	97.5	A+	Best Imaginable	Acceptable

Table 9.7: SUS score for each NAV employees alongside the corresponding grade, adjective and whether or not the result is acceptable

This gives an average SUS score of:

$$\overline{SUS} = \frac{70.0 + 87.5 + 95.0 + 80.0 + 97.5}{5} = \frac{430}{5} = 86.0$$

Average SUS Score Throughout the Third Iteration.

Total average score for the third iteration:

$$\overline{SUS} = \frac{81.5 + 86.0}{2} = \frac{167.5}{2} = 83.75$$

Chapter 10

Final Changes

In this chapter, any final changes made to the carpenter app will be described. Normally this would be the 4th iteration, but since there were no more opportunities to perform user tests, there are no results for the changes made in this iteration. In addition, relatively few changes were made between the third iteration and the final product. The main change is regarding navigation, which was a problem outlined in the feedback from the third iteration (9.2.2).

10.1 Changes

10.1.1 Waypoints

Results from the previous iteration show an apparent lack of information about where the user needs to go to the next task, both for the tutorial and the main application. Waypoints and arrows were added at key locations in the app to fix this issue. The tutorial rooms now have arrows on the walls which show which direction to go. Doors also have exit signs above them, so there is no doubt which door the user must go through. The main app needed something similar, but since it isn't cramped like the rooms in the tutorial, having small arrows on the wall would probably not suffice. Instead, large waypoints showing the way were implemented and added for the wall and frame task. These waypoints are similar to those found in the windmill app (3.1.2), which are essentially large cylindrical arrows floating in the air. A key difference is that the arrows added to the carpenter app were a bit more protruding. In addition to adding a visible marker on the ground, the arrow has an animation where it moves up and down indefinitely. The idea here is that it would be easier to spot it, and there should be virtually impossible for users to miss it once they advance to the main game. Furthermore, these waypoints will disappear once the user teleports to the task so that they don't obstruct visibility on the workstation.

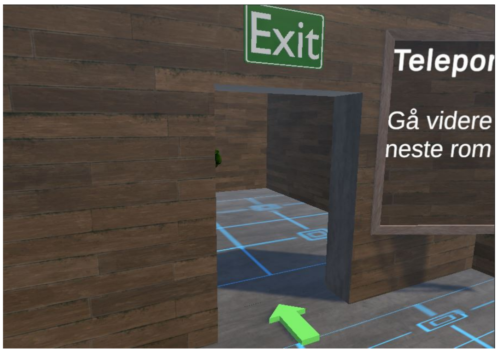


Figure 10.1: Door with exit sign and arrow indicating the way.

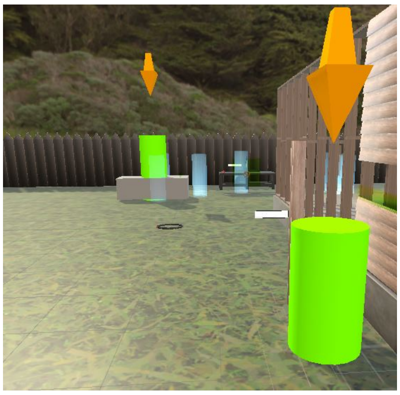


Figure 10.2: Waypoints showing where the task is.

10.1.2 Feedback When Performing Tasks

Another frequent observation was that users would not know whether or not they had completed their task. Once again, inspiration was taken from earlier Immersive Job Taste apps to remedy this issue. A common solution used by previous apps is to play sounds when the user does something wrong or if they have completed a task. When a player completes a task or a subtask, a distinct sound will play depending on what the user did. If the user completes a single task step, a brief pling sound will play, indicating that the step was done correctly. If the user does something incorrect such as cutting a plank too short or choosing the wrong plank type, the app will play an error sound.

Additionally, there is textual feedback if the user does something wrong, stating what they have done wrong and what they need to do. For instance, if the user cuts a plank too short, the board for that task will show red text stating that the plank they cut is too short. Finally, if users complete a task, a success sound will play. The success sound is very different from the pling sound played when completing a step, and it lasts much longer to indicate that the entire task has been finished.

10.1.3 Ironing Out Bugs

Testing revealed a couple of bugs in the app that were fixed. One of the said bugs was being able to grab things from far away. This was likely due to long-distance grabbing, which was experimented with a while back but was ultimately removed since it did not work as intended. Somehow the distance grab had not been removed completely, which is why there were instances of players grabbing planks on the ground by accident. Removing long-distance grab seems to have fixed the issue.

The circle saw had two bugs, one found through user testing and one discovered during development. A user found a bug where the snap zone on the saw would be placed incorrectly. Planks would be snapped incorrectly as a cause of this making it impossible to cut the plank to the appropriate length. This was fixed by tweaking how the cutting block calculates where to place the snap zone based on the length of the plank. The second bug was related to rotating the circle saw. In all the previous iterations, the saw would only work correctly if its rotation was set to 0 on the y-axis. This was technically not a problem as the circle saw was positioned in the desired location, but it is something that could pose a problem in the future. The issue lies in how the circle saw calculates the move direction when moving the cutting block. Specifically, it was due to calculating incorrect positions for the start and end of the move direction. The move direction used to tell the cutting block which way to move is calculated by creating a vector. This vector is made from two positions on the circle saw. Due to erroneous math, one of these points was placed in the wrong location. After rectifying the mistake, the circle saw could be used from any rotation on the y-axis.

A final bug discovered when developing was that if the framing task were completed first, it would also satisfy the wall task since the same requirements would be fulfilled – nailing three planks together. Fixing this was a simple task, as the app must check which task the user is currently doing. This is done by having an internal state which tells whether or not the wall task is activated. This state is updated when teleporting to the different workstations. If the user is currently doing the frame task, the wall task manager will not update its progress since the wall task is not set to active.

Chapter 11

Discussion

11.1 Limitations

One of the limitations of this project was time, resources, and what path to follow. Although two students are working on the master thesis, there is much overlap between the competence they possess. While there were areas where one excelled better than the other, for the most part, the knowledge of programming and using Unity was comparable (see chapter 4.2). This means there was much trial and error to get things to work, especially technologies such as VRTK. It was also not entirely clear initially what this master thesis was supposed to focus on. Lots of discussions led to different topics and thus different research questions. Initially, the focus on multiplayer seemed desirable but was later changed to something else. One of the students underwent courses focused on designing the graphical user interface and participatory design, which entails how one should develop products with the end-user. Our supervisor suggested this as a point of research since it was known beforehand that the UI in the existing job taste apps could be improved. Eventually, the final consensus was that not only UI but usability in general for job taste apps would be the main focus of this thesis.

The results are also limited due to the relatively few users that would test the app when comparing the number of users other Immersive Job Taste projects received. This point is further underlined by the fact that not all users decided to answer the questionnaire in the second and third iterations. The first iteration did not have a questionnaire, another limitation of the results. It would be interesting to see how presumably poor the usability was of the first iteration. Still, since there is no data for this, only speculations from the observations can be drawn, which are not as credible as raw numbers.

A limitation of developing the carpenter app was access to reliable information about the carpenter profession. Early on in the project, it was decided that a company with expertise in carpentry would be contacted to receive credible information regarding everyday tasks done by a carpenter. We had sent out some questions about the profession, some simple tasks for trainees, and essential skills. Unfortunately, contact with the said company was never established due to several delays, and ultimately no response was received. Consequently, all tasks in the carpenter app are made by taking inspiration from whatever information is available about the profession, such as web pages and YouTube videos.

11.2 Research Questions

This chapter will discuss the results that were found during the case study and user testing of CarpenterVR. The research questions will be discussed in detail and great length. Shorter and more concise answers can be found in the conclusion (section 12.2).

RQ1: How to improve System Usability with UI in Immersive Job Taste apps?

Based on the results achieved throughout the different iterations, it became clear that the best UI was the one that was not used. This might seem paradoxical, but user testing revealed that most people did not use the menu or the tablet nor the new version described in 7.3.6. This led to situations that would impede the playability of the app. Since users were unfamiliar with the controls initially, it led to numerous instances where they would open the menu by accident and did not know how to get rid of it. Sometimes this would happen at a workstation which meant the menu would obscure objects such as planks and tools, preventing the user from progressing before closing the menu.

The accidental appearance of the UI in our application caused the user to struggle with how to close the UI. This would most likely break immersion. These problems are echoed in the VR UI Guidelines chapter 3.5. Interviews from the paper pointed out that UI often breaks immersion and that UIs are often confusing to access and control[26].

This was not true for all our users, as there were instances where some of them understood the concept of having a menu that pops up and explains the different tasks. That said, they were a minority, and the vast majority of the time, users would be noted its existence, try it out for the first task and then never use it again. However, not using the menu could be due to a few reasons. Firstly, many users were very eager to try out the different tasks, making it easy to forget about the less interesting menu that contains mostly walls of text. Secondly, it could also be because most users had little VR experience (68% of users had none or little experience with VR) and are therefore not familiarized enough with the controls or how VR games work. Whatever the reason, having a menu that needs to be opened to understand the task from the users' side seems to cause the menu to end up being unused. Thus, creating UIs that do not need to be constantly opened by the user to complete tasks is better. The Immersive Job Taste methodology uses 360 videos to describe tasks, but this does not entirely translate one-to-one in the apps. Incorporating the GUI into the user's environment is one of the few guidelines in all 3 of Oculus, Unity, and Leap Motion's guidelines for creating GUIs in VR [26]. By developing tasks to be understandable in the VR environment, the user does not need to open the UI to understand how to do the task. This circumvents the need to have a UI for this purpose as UI is often used for this purpose in the previous Immersive Job Taste apps[6][2][5]. The developer can decide what information is relevant for the user based on the user's location. The user does not need to guess what task they will do and navigate the UI to understand what they need to do. This removes complexity from the game and only makes the game less tedious. Once the UI was no longer a part of the carpenter app, the users had fewer questions about the tasks they were supposed to do, and less guidance was needed from an external supervisor. This might increase the usability for young job seekers and supervisors as it takes less supervision to use the application.

On the other hand, not enough testing was done to disregard interactable menus from Immersive Job Taste apps. However, for the UIs used in these apps, it is strictly unnecessary to centralize all tasks into one UI. Ultimately the purpose of the UIs in the previous Immersive Job Taste apps is to convey information about the tasks and the user's score. This happens through reading text explaining various game concepts, such as how to perform specific tasks or how many points they received by completing one. In the end, it is not essential that the user can interact with the UI in some shape or form. Instead, it creates a better flow in the game if the user is given the information they need at the appropriate time and location instead of having to actively open a menu as this is immersion breaking 3.5.

In chapter 9.2.1 it was observed that many users would read instructions on what to do several times to complete the task. This is to be expected since there are usually several steps to complete a task; thus, it is challenging to remember every step after reading the instructions. Therefore, having an interactable menu that pops up with a button press would lead to a poor user experience since the menu would have to be opened several times. Each time, the user would have to navigate to the task description to read the next step. Describing the tasks in the virtual environment removes the hassle of constantly opening and closing the UI in-between steps. Instead, the user can turn their head to face the UI when pondering the next step or if they forgot the current one. This removes the complexity of some of the game and makes it more streamlined as the user only needs to focus on the tasks and not where to find information about how to do the task.

The boards in the carpenter app took the job of the menu that the previous Immersive Job Taste apps had. The boards follow many UI guidelines, even if it is not very interactable. However, it is intractable to some extent. The interaction is based on the progression of the task, not interactive elements on the board. The boards follow many UI guidelines in chapter 5.1. The guidelines based on interactive elements are hard to rate as the boards do not have the interactive elements on them. The UI implemented in the first iteration did follow all guidelines except "**Use wearable menus**".

Guidelines	**CarpenterVR Board System**	**CarpenterVR first iteration menu system**
Comfortable content distance 0,5m -1m	Partially	Yes
Incorporate the GUI into the user's environment or character	Yes	Yes
Use texts in UI that are easily read	Yes	Yes
GUI surrounding the user	No	Yes
Provide visual feedback on interactive elements	*	Yes
Use wearable menus	No	No
Scale and spacing suitable for interactive elements	*	Yes
Avoid pinning GUI in the user's view	Yes	Yes
Avoid content in peripheral areas	Yes	Yes
Prevent the virtual hand from obscuring interactive elements	Yes	Yes

Table 11.1: * The Board system do not have interactive elements on the board

RQ2: What value does the carpenter app provide as an Immersive Job Taste?

When discussing the value of the carpenter app, it is important to highlight its strength and weaknesses. First off, it fulfills the primary purpose of this master thesis, which is to determine ways to increase the usability of the Immersive Job Taste apps. In that regard, it performs well by receiving a SUS score of 83.75 on average. Some of the previous Immersive Job Taste apps performed noticeably worse in this regard, such as the car mechanic app receiving scores ranging from 63-65 on average with considerable discrepancies in the scores recorded, with some going as low as 32.5 [5]. Other apps, such as the crane app and mining app, received a score of 69 and 70 respectively[6][43]. These results indicate that the carpenter app is more user-friendly and can serve as an example for other job taste apps regarding what could be improved. More testing would be required to be certain as some of these SUS scores are calculated based on a small sample of user data.

On the other hand, having ten users on the third iteration is not as insufficient of a number as one might imagine. As five can often be a magic number of answers for a SUS score, this amount gives a surprisingly stable average score which is often close to the actual SUS score [44]. As the third iteration has double the minimum number of SUS scores it is probably close to the correct SUS score.

Another value of the carpenter app is its usefulness for NAV as a tool to engage young job seekers. Its usability is already established, but what about the insight it provides about the carpenter profession? This would be the area where it is a bit lacking compared to other job taste apps. The windmill app received a slightly better score on the questionnaire, giving the user a better understanding of the job. Questions such as “This type of application can give me a better understanding of the workplace” or “This application gave me insight into how I should complete certain tasks at the workplace” regularly received a score of 4 or higher on the Likert scale. In contrast, the carpenter app received four or lower. The disparity between the apps is noticeable and not too surprising. For one, the carpenter app focused more on usability than providing insight into the profession. Secondly, the tasks in the carpenter app were made based on what tasks could be found on the web instead of asking carpenters what everyday tasks are done by trainees. Thus, the tasks in the carpenter app might not wholly represent the main tasks that a carpenter does. It also lacks in quantity when it comes to tasks. This is also due to the tasks, on average, being slightly more complex than tasks in other apps. Regardless, the carpenter does give some valuable insight into the carpenter profession. However, there is much room for improvement, such as adding more tasks and validating the tasks already made to ensure they accurately represent the profession. Another point could be that the carpenter app is one of the only apps that does not have 360 videos, potentially an essential aspect for receiving insight into the profession. This makes the carpenter app one of the apps that do not fully follow the methodology of Immersive Job Taste methodology [1].

Another value of the carpenter app is the tutorial that comes with it. Whether or not it improves usability is debatable, but it is evident that it enables users to be more self-sufficient in using the app. This can be of great value as it is technically unnecessary for a supervisor to explain how the app works to the user. It makes it possible for users to play the app from home if they have a VR headset, a device that has become much more accessible in the last decades. This could skip the step where the user must meet up at NAV with a supervisor. Unfortunately, this would eliminate the part where the young job seeker discusses their options with the supervisor, as the Immersive Job Tastes usually work as a conversation starter. The tutorial still adds value even if used alongside a supervisor, as it removes the need for the supervisor to explain how to use the app. This is beneficial for several reasons. One of them is that it is easier to explain how the game works by showing text and animations instead of trying to explain through words. As the Immersive Job Taste catalog increases, the supervisor might not remember how all Immersive Job Taste apps works, which is another value the tutorial provides.

RQ3: What guidelines should be considered when designing Immersive Job Taste apps to enhance the user experience?

When designing Immersive Job Taste apps, there are a few things to consider. First and foremost are the end-users that will use the app. The target audience of the Immersive Job Taste apps is young job seekers such as school students or people that are unemployed [1]. Developers should therefore assume that the end-users are people without much work-life experience. With this in mind, it is essential that the Immersive Job Taste apps are understandable and do not use professional language without explaining its meaning.

The questionnaires in B.1 also revealed that most users are novices regarding VR, meaning that a large portion of users has zero experience in using VR. As such, it is crucial that the app is simple to use and understandable to include all users regardless of their experience with VR and video games. A precaution to consider on this topic is making simple controls and a tutorial alongside it. Both aspects are important guidelines one should follow when improving the user experience [45][46]. This is also reflected across the different iterations throughout the project. The second and third iterations had a 4.4 average Likert scale on the question "It was easy to understand what I was supposed to do in the carpenter app." It shows that most users find it easy to understand the application.

In the beginning, the control scheme was cluttered with lots of different functionality that was not strictly needed, such as having two movement systems and mapping the menu to several buttons on the controller. On top of that, there was no tutorial in the first two iterations meaning that all of this had to be explained to the user by a supervisor. The observations from table 7.4.1 clearly show users' numerous problems due to the app not being simple enough. As already mentioned in RQ1, the menu would be opened a lot by accident, obstructing gameplay. This is likely due to having the menu mapped to several buttons, which increases the chance of unintentionally opening it. If menus are added to Immersive Job Taste apps, they should not be easy to open by accident. Suggestions here would be to map the menu to obscure buttons on the controller that is not as easily reached and require the user to hold down the button for it to register. Having the menu on an obscure button might also not be the best solution, as if the user opens the menu by accident, it will be harder to close it with the same button. If UIs are implemented, it is essential that it is easy and understandable to close them. The movement should also be as rudimentary as possible not to break the flow of the game. One way to implement a control scheme is to use *natural mapping* (2.4.4). One of the tools used in the carpenter app is a nail gun. The first control scheme had the user use the trigger button to fire the nail gun and the grip button to grab objects. This meant the user needed to hold the nail gun with the grip button and fire it with the trigger button. This is close to how a real nail gun would operate. The person grips the nail gun with the middle finger and pulls the trigger with the index finger. This caused problems as users did not see or feel the grip button and did not know its existence. Using *signifiers* might have solved this problem, but as it looked like most users tried to press the buttons on top of the controller to fire the nail gun, it was changed to use the buttons on top. The controls were simplified substantially, and the grip button was no longer used as it was hard for users to discover its existence (2.4.2).

Some users could not accurately control the teleport pointer, which caused many players to teleport to undesirable locations. This issue was particularly prominent in the first iteration as users were allowed to teleport virtually anywhere on the map. As such, teleportation benefits from the concept of *constraints* (2.4.6), where the user is physically constrained to teleport to certain locations. These *constraints* do have *signifiers* showing the constraints as well as visual *feedback* when teleporting (2.4.2, 2.4.5). In addition to limiting the space where users can teleport, it is also beneficial to have snapping on the teleport laser. This mitigates the problem of users overshooting or undershooting when trying to teleport to the point of interest, such as a workstation. This is also reflected in the results, as when snapping was introduced in the second iteration, the issue was no longer present for all intents and purposes. Users would no longer teleport far off into the distance due to poor aim with the controller.

The difficulty of tasks is also a topic of concern one should evaluate when designing tasks in the job taste apps. The case study (chapter 5) and user testing for the carpenter app show that some of the tasks in Immersive Job Taste apps are unnecessarily complicated. Apps such as the tinsmith and car mechanic app had some tasks that were difficult to do or figure out even for someone experienced in VR. However, many of the tasks in these apps are difficult because the apps do not explain to the user how to do the task. They have explanatory 360 videos on the task; however, these are not always sufficient if the user does not understand how the controls work. In other words, tasks should not only be explained how to be done in the real world but also how to do them in-game. Knowing how to do a task in the real world means nothing if the user does not know which button(s) to press to perform the action in-game.

The way the tasks are described in the carpenter application is to have the description of the task next to the task in the environment. There exist more subtle forms of guiding the user through the tasks. Demonstrating how to make the task motions with a transparent “ghost” gives the user self-efficacy[45]. This shows how to do the task correctly without the need for text. This might be a better way of demonstrating the task, or they could be combined to have two ways of displaying the task.

In some instances, the problem with the task might not be that the user does not understand how to do the task, but the mechanics of the game makes the tasks impossible. A prime example of this was the wall nail task in the carpenter app's first iteration. The difficulty arose due to 2 main reasons: precise placement and tool manipulation. Planks were difficult to place since they did not have snap zones in the first iteration. The nail gun was also not attached to the belt, meaning users had to carry it while moving to the wall task. A common issue was that the nail gun would be dropped a lot by accident, so the users had to spend time trying to retrieve it. On top of that, the nail gun could undo progress by knocking planks off the wall. Most users would not have this big problem in the real world. This is mainly due to the limitations of VR. One can feel how the planks press against the wall in real life, but in VR, it does not feel like anything. The weight of a nail gun on a plank helps remind the user that they need to hold on to it. This feedback is not possible in VR and makes the tasks much harder. Solutions to these kinds of problems are to make mistakes easily correctable.

An example would be mechanisms that prevent the user from losing essential items in the game, such as the nail gun in the carpenter app. In the first iteration, the nail gun was a loose object that could be picked up and dropped anywhere, but in the second iteration, it would always return to the toolbelt if the user dropped it. This ensured that users would not start frantic rescue missions whenever they lost the nail gun.

A great point of discussion is how difficult Immersive Job Taste apps should be while preserving a good user experience. It is clear from the empirical examples that tasks need to be adequately explained when it comes to how to do the task while also knowing the controls. Tasks should not be frustratingly difficult; thus, it is best to avoid tasks that require precision by the user. This is because it is difficult to perceive things such as depth and make precise movements using the VR controllers. On the other hand, tasks should not be too easy as that would diminish the engagement and insight that the apps provide. The tasks should replicate their real-life counterpart as accurately as possible without being too technical while also introducing leniency so that the user does not need to be an expert at using VR to perform the tasks. It is essential to base the task's difficulty based on the user's competence. Most of the Immersive Job Taste apps are short VR experiences. The time to build competence in the app is short. Thus, the difficulty should not be very high[47]. It is advisable to have elements such as snap zones and things of similar nature that helps the user in completing the task.

During the third iteration, a tutorial was added to the final product and became a prerequisite before allowing the users into the main app. The addition of this tutorial seemed to benefit the user experience notably. According to a study on the guidelines in VR play [46], some of the most impactful guidelines were E5 **integ-**

rate tutorials with gameplay and E6 **teaching mechanics one thing at a time** (stepwise learning). These were deemed important guidelines to follow when it comes to improving the player experience in VR games. As the tutorial is meant to build competence in the VR app, the VR app can have harder difficulty. A better understanding of the application and challenging enough tasks engage the user (positively excited)[47]. Also, having a good tutorial makes it less likely that the user is negatively excited (anxiety) due to having some competence from the tutorial[47]. The results for the carpenter app show similar results. Firstly, the SUS scores improved significantly, going from a good 75.3 (B) to an excellent 83.75 (A) between the second and third iteration when the tutorial was added. As discussed in the limitations, these results should be handled carefully as there were not as many answers received in the third iteration. It is also impossible to directly correlate the increase in SUS score to a particular change made in the game when several changes were made. Despite this, it is apparent that the tutorial had a good impact on the playability of the carpenter app. During user testing, it became clear to see that some users were able to complete the task in its entirety without guidance from a supervisor. This means that the tutorial can explain all the concepts in the app and make users self-sufficient in using the app. Some questions were still asked, but there were noticeably fewer, which indicates that the tutorial works as intended.

Adding a tutorial is an important precaution one should take when designing and creating an Immersive Job Taste app. The vast majority of the previous apps made do not have a tutorial. In the case study, only one of the apps had a tutorial: the windmill app (5.3.4). However, the tutorial is not a complete one as it only teaches how to teleport. The tutorial does not explain other mechanics such as grabbing, opening, and navigating the menu or how to climb the ladder inside the windmill. In all fairness, this is explained during gameplay as the user explores the app. Further work could include comparing these two different tutorial systems to see which performs better. In any case, having a tutorial is something the previous and future job taste apps would benefit immensely from.

11.3 Requirements

At the beginning of the iterations, requirements for the carpenter app were established. In this section it will be discussed which requirements that were fulfilled, and which become the basis for future work. Table 11.3 highlights all the requirements and to what degree they were achieved. As done in chapter 5, the cells in the table contains either *Yes*, *No*, or *Partially*. Yes and No means the requirement was completely fulfilled or not at all respectively. Partially means that the requirement was fulfilled to some degree, but not completely, and as such should be improved upon as future work.

Requirement	**Fulfilled?**
The application has to be compatible with the Oculus Quest HMDs	Yes
Tasks should be added where the user can cut materials and nail together items	Yes
A tutorial needs to be added in order to explain the game	Yes
The application needs a UI that follows the guidelines found in the literature review	Partially
The application needs to provide scores in order to evaluate the user's performance	Partially
The application should give feedback to the player when they do tasks to guide them	Partially
The application needs to have a sufficient FPS preferably 90 or more	Partially
The application should use VRTK for quick development	Yes
The tutorial needs animations or "ghosts" to show what to do	Partially
The tutorial needs to teach users concepts stepwise	Yes
The application needs to be simple so that all users can complete the tasks	Partially
The application needs to support several languages	Yes

About half of the requirements were achieved completely while the other half was mostly achieved. The UI guidelines from 3.5 were followed for the most part with a couple of exceptions. The first iteration followed every guideline with the exception of using wearable menus. The final version lacked in addition to wearable menus, a GUI that surrounds the user. The task boards are completely flat as they are rectangles, so in order to follow this guideline, the board would have to be curved. Whether or not the carpenter app needs to follow all the guidelines is arguable, since the boards do not benefit from being wearable as they are too big. Having curved boards also might not help all that much when they are placed relatively far away from the user.

The final application does have scores which evaluates the user's performance; however they are fairly one dimensional. You can either get 0 or 100 points, which means that every user that completes the task performs equally well according to the system. The first iteration did have varying scores for the cutting task depending on the length that was cut, but since this task was reworked, the final version does not take this into account. Future work in this regard could be to evaluate the tasks based on how well they are done. For instance, the frame task could evaluate the length of the planks that were cut and nailed together, or the wall task could check if the planks were nailed in the correct spots. Feedback is another requirement that was partially achieved. All tasks plays a sound whenever they are completed, but the wall task in particular does not give any other feedback besides that. A suggestion here would be to give the user some feedback when they have nailed one plank to the wall by either creating a sound and/or showing their progress on the task board. The frame task gives both sound and visual feedback in the form of text whenever the user does something wrong as explained in section 10.1.2.

A tutorial was successfully created and added to the app and appears to be working as intended. Users were able to become more self sufficient and most did not require guidance to figure out how to use the application. Still, the animations in the tutorial are lacking compared to what they could be, which is why this requirement was only partially achieved. Currently there are Oculus controllers with an animation showing which button to press, however this only works for showing the controls, but not how to actually perform the tasks themselves. The only reason users know what to do is because the task boards contains text with a thorough explanation on what to do. Regardless, it would help massively if there were some animations that would show the users how the task should be performed. Previous job taste apps have 360 videos which lets the user see how the task is supposed to be done in real life. Having this could augment our tutorial as a 360 video is likely to be more engaging for users than to read text on the wall. Alternatively a “ghost” or 3D avatar could be modelled and animated to show the user what to do which is a guideline found in VR Play that was found to be very impactful for the user experience [46].

Chapter 12

Conclusion and Future Work

12.1 Conclusion

This master thesis has looked at improving usability in Immersive Job Taste apps. A literature review has been done to determine what guidelines should be followed when designing the UI for these apps. Said guidelines were then used to test prior job taste apps UIs. Results show that it varies from app to app, but the guidelines are mostly followed either entirely or to some extent. Some guidelines were more frequently violated than others, such as avoiding pinning GUI in the user's view or having the GUI surround the user.

The knowledge gained from the case study and the guidelines established during the literature review were then utilized to design and create an entirely new Immersive Job Taste app for the carpenter profession. A tutorial was also made specifically for the carpenter app, which teaches the user how to use the app. Results showed that although the tutorial does not necessarily increase usability, it made users more independent and self-sufficient when using the app. It also showed that simplicity is key to making these apps provide a good user experience. Tasks should not be overly challenging, and the controls need to be simple for a smooth user experience. When designing a UI, one should consider whether or not the information in the UI should be in the UI or the environment. If all the UI does is display text or a list of actions on how to do tasks. There is no need for it to be interactable, as results show that users are unlikely to use tablets and similar menus to retrieve information about tasks. The final product is something that adds value both as an Immersive Job Taste app that can engage young job seekers to a moderate degree while also working as a solid example of how job taste apps should be designed regarding usability.

12.2 Research Questions

RQ1: How to improve System Usability with UI in Immersive Job Taste apps?

The menus of Immersive Job Taste might not be necessary for many applications. The use of menus can be confusing and immersion-breaking. Most of their value in other applications can be transferred to the user's environment. Things such as points and explanations can easily be transferred to the user's environment, making them easier to access and removing the complexity of navigating the menu. There might be some value to having a menu, but having a menu that is immersion-breaking and users need to learn in a short VR experience might not improve the system usability of the application. If UIs should be used, they should be simple, not be necessary to get the whole experience of the job tasks, and implemented so that they don't break immersion and don't open accidentally.

RQ2: What value does the carpenter app provide as an Immersive Job Taste?

1. An Immersive Job Taste app with high System Usability Score
2. An example to other Immersive Job Taste apps to improve System Usability
3. A tool to engage and empower young job seekers for NA
4. A tutorial to explain how the application works and as an example for new Immersive Job Taste apps

RQ3: What guidelines should be considered when designing Immersive Job Taste apps to enhance the user experience?

These are the guidelines we recommend:

1. **Make a Tutorial for the game's mechanics.** Tutorials are important to build the user's competence and confidence. They are making users understand the mechanics and controls of the game.
2. **Make the game simple.** The Immersive Job Taste apps are short VR experiences. This means the user does not have enough time to learn complicated tasks. The app needs to build competence in the user for more complex tasks. Because the VR experiences in Immersive Job Tase are short and to make them as accessible as possible, it is essential to make the games simple.
3. **Make the control scheme as simple as possible.** This might take trial and error to accomplish. The controls should be similar to previous Job Taste apps, but this is not always possible. Making the controls as easy as possible will make it easier for the user to do the different tasks. The controls should be explained in the tutorial as well.
4. **Make menus very easy to close.** The menu should be easy to close as unintentional openings can cause the users to become confused and break immersion.

5. **Avoid using professional language without explaining its meaning.** This is to make the VR experience as accessible as possible. If the tasks are defined using professional language that the user does not understand, then the task is inaccessible to that user.
6. **Avoid using a menus to explain the work tasks.** UIs like menu or alike should not be used to describe tasks. This reduces the user's need to navigate the menus to complete tasks. Instead, use the environment to describe the tasks. Using the environment to explain what to do makes finding information more accessible.
7. **Accuracy and precision tasks should implement mechanics so the user has good control.** As precision can be challenging for new VR players, and mechanics should be implemented to counteract and give more control to users.
8. **Essential tools or game objects should be impossible to lose.** Game objects were often dropped in the carpenter app. This caused the users to try to pick them up while not hitting the floor in the real world. Also, essential items could be lost, thus making some tasks impossible to complete.

12.3 Contributions

Listed below are the contributions that the carpenter app and this thesis bring to the Virtual Internship project:

- **A new Immersive Job Taste app:** A new job taste app that can be used for engaging young job seekers as well as used or for further research such as expanding the app with new tasks or validating the claims about its usability.
- **Guidelines for improving system usability in Immersive Job Taste apps:** This thesis includes some guidelines for creating good UIs in VR as well as some key elements that should be present for gameplay to be as user-friendly as possible.
- **Components which could be used in other existing apps:** Systems such as the task or language manager can be used in other apps to make them more accessible. Language especially is important as many international users test VR apps in the IMTEL lab, and it would be beneficial to include support for additional languages.

12.4 Future Work

This section highlights aspects of the app or thesis which can be further developed and improved.

12.4.1 Testing the Newest Iteration

After the third iteration, some changes were made to improve the app based on the feedback received. These new changes were not tested on users as there was no time left. It would be interesting to see if this new version ranks as high, if not higher, than the ones from previous iterations. The second and third iterations received a SUS score of 75.3 and 83.75, respectively, corresponding to the grades B and A. It would be interesting to see if it would be possible to make the app even further and receive a SUS score in the 85 range, which would be equivalent to an A+, which is only achieved by the top 5% of all systems[8].

12.4.2 Adding New Tasks and Validating Existing Ones

Currently, the app only consists of two tasks for the carpenter profession alongside a tutorial. The next step should therefore be to validate that these two existing tasks accurately represent the carpenter profession as information about the tasks was gathered from the web. Validation should ideally be done by contacting carpenters for a reliable source of information. Additionally, new tasks should be added as the app is shorter than most Immersive Job Taste apps due to the small map and few tasks.

12.4.3 Applying Principles to Existing or Future Immersive Job Taste Apps

The carpenter app was designed with usability in mind for UI and gameplay by following guidelines. This could be taken further by investigating if the results achieved apply to other job taste apps that did not get relatively as high of a SUS score. For example, the car mechanic received a relatively poor score of 63-65, giving it a C or C-. One could, for example, try to remove the interactable tablet and replace it with a system similar to the board system made for the carpenter app to see if it yields higher usability. Likewise, a tutorial and better explanations of the different workstations could be added. This would validate whether or not the results achieved are in tandem with the literature review that was done.

Bibliography

[1] M. Fominykh and E. Prasolova-Førland, 'Immersive job taste: A concept of demonstrating workplaces with virtual reality,' in *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, 2019, pp. 1600–1605. DOI: 10.1109/VR.2019.8798356.

[2] J. Henrichsen, 'Engaging young job seekers with an internship as a wind turbine technician in virtual reality,' M.S. thesis, Norwegian University of Science and Technology, Høgskoleringen 1, Trondheim, Nov. 2019.

[3] 'Virtual internship.' (2022), [Online]. Available: https://www.ntnu.edu/imtel/virtual-internship. (accessed: 05.06.2021).

[4] K. H. Håkonsen, E. H. Aasland, A. Madslien, A. Haukås and J. Ness, 'Prosjektrapport eksperter i team,' NTNU, 2021.

[5] S. Ulvestad, 'Career guidance in collaborative virtual reality,' M.S. thesis, Norwegian University of Science and Technology, Høgskoleringen 1, Trondheim, Sep. 2021.

[6] R. E. Maikher, 'How to create accessible virtual reality (vr) experiences to aid young job seekers with career guidance,' M.S. thesis, Norwegian University of Science and Technology, Høgskoleringen 1, Trondheim, Nov. 2021.

[7] *Unemployment rate after the coronavirus outbreak in norway in 2021, by industry*, https://www.statista.com/statistics/1113595/unemployment-rate-after-the-coronavirus-outbreak-in-norway-by-occupation/, Accessed: 2022-04-01.

[8] J. Sauro. '5 ways to interpret a sus score.' (), [Online]. Available: https://measuringu.com/interpret-sus-score/. (accessed: 30.05.2022).

[9] S. M. LaValle, *Virtual Reality*. Cambridge, CB: Cambridge University Press, 2020.

[10] E. A.-L. Lee and K. W. Wong, 'A review of using virtual reality for learning,' in *Transactions on Edutainment I,* Z. Pan, A. D. Cheok, W. Müller and A. El Rhalibi, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 231–241, ISBN: 978-3-540-69744-2. DOI: 10.1007/978-3-540-69744-2_18. [Online]. Available: https://doi.org/10.1007/978-3-540-69744-2_18.

[11] D. Allison and L. F. Hodges, 'Virtual reality for education?' In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology*, ser. VRST '00, Seoul, Korea: Association for Computing Machinery, 2000, pp. 160–165, ISBN: 1581133162. DOI: 10.1145/502390.502420. [Online]. Available: https://doi.org/10.1145/502390.502420.

[12] D. Allcoat and von MühlenenAdrian, 'Learning in virtual reality: Effects on performance, emotion and engagement,' *Research in Learning Technology*, vol. 26, Nov. 2018. DOI: 10.25304/rlt.v26.2140. [Online]. Available: https://journal.alt.ac.uk/index.php/rlt/article/view/2140.

[13] K. at Wharton Staff. 'Can the minerva model of learning disrupt higher education?' (2018), [Online]. Available: https://knowledge.wharton.upenn.edu/article/can-the-minerva-model-of-learning-disrupt-higher-education/ (visited on 04/06/2022).

[14] V. R. LeBlanc, A. Urbankova, F. Hadavi and R. M. Lichtenthal, 'A preliminary study in using virtual reality to train dental students,' *Journal of Dental Education*, vol. 68, no. 3, pp. 378–383, 2004. DOI: https://doi.org/10.1002/j.0022-0337.2004.68.3.tb03754.x. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/j.0022-0337.2004.68.3.tb03754.x. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/j.0022-0337.2004.68.3.tb03754.x.

[15] M. Zyda, 'From visual simulation to virtual reality to games,' *Computer*, vol. 38, no. 9, pp. 25–32, 2005. DOI: 10.1109/MC.2005.297.

[16] A. Smith. 'The priesthood at play: Computer games in the 1950s.' (2014), [Online]. Available: https://videogamehistorian.wordpress.com/2014/01/22/the-priesthood-at-play-computer-games-in-the-1950s/. (accessed: 07.012.2021).

[17] A. Smith. '1tl200: A magnavox odyssey.' (), [Online]. Available: https://videogamehistorian.wordpress.com/2015/11/16/1tl200-a-magnavox-odyssey/. (accessed: 07.012.2021).

[18] videogamehistorian. 'Nintendo entertainment system.' (2015), [Online]. Available: https://videogamehistorian.wordpress.com/2015/11/16/1tl200-a-magnavox-odyssey/. (accessed: 07.12.2021).

[19] S. Retro. 'Sega master system/technical specifications.' (), [Online]. Available: https://segaretro.org/Sega_Master_System/Technical_specifications. (accessed: 08.06.2021).

[20] 'Ergonomics of human-system interaction — Part 210: Human-centred design for interactive systems,' International Organization for Standardization, Geneva, CH, Standard, Jul. 2019.

[21] D. Norman, *The DESIGN of EVERYDAY THINGS*. New York, NY: Basic Books, 2013.

[22] J. Brooke, 'Sus: A quick and dirty usability scale,' *Usability Eval. Ind.*, vol. 189, Nov. 1995.

[23] C. L. VR. 'About us.' (2022), [Online]. Available: `https://careerlabsvr.com/about-us/` (visited on 17/04/2022).

[24] Z. H. Sim, Y. Chook, M. A. Hakim, W. N. Lim and K. M. Yap, 'Design of virtual reality simulation-based safety training workshop,' in *2019 IEEE International Symposium on Haptic, Audio and Visual Environments and Games (HAVE)*, 2019, pp. 1–6. DOI: `10.1109/HAVE.2019.8921366`.

[25] J. Yang, F. Liu, J. Wang, Z. Kou, A. Zhu and D. Yao, 'Effect of virtual reality technology on the teaching of urban railway vehicle engineering,' *Computer Applications in Engineering Education*, vol. 29, no. 5, pp. 1163–1175, DOI: `https://doi.org/10.1002/cae.22371`. eprint: `https://onlinelibrary.wiley.com/doi/pdf/10.1002/cae.22371`. [Online]. Available: `https://onlinelibrary.wiley.com/doi/abs/10.1002/cae.22371`.

[26] S. Alves, A. Callado and P. Jucá, 'Evaluation of graphical user interfaces guidelines for virtual reality games,' in *2020 19th Brazilian Symposium on Computer Games and Digital Entertainment (SBGames)*, 2020, pp. 71–79. DOI: `10.1109/SBGames51465.2020.00020`.

[27] *Vr best practices guidelines*, 2015. [Online]. Available: `https://developer-archive.leapmotion.com/assets/Leap%5C%20Motion%5C%20VR%5C%20Best%5C%20Practices%5C%20Guidelines.pdf`.

[28] H. H. Kvalnes, 'Immersive interface in virtual reality,' M.S. thesis, Norwegian University of Science and Technology, Høgskoleringen 1, Trondheim, Sep. 2021.

[29] B. J. Oates, *Researching Information Systems and Computing*. New York, NY: SAGE Publications Ltd, 2006.

[30] Unity. 'Unity.' (2015), [Online]. Available: `https://unity.com/`. (accessed: 27.05.2022).

[31] Unity. 'Unityarchitecture.' (2022), [Online]. Available: `https://unity.com/solutions/architecture-engineering-construction`. (accessed: 27.05.2022).

[32] Unity. 'Unityautomotive.' (2022), [Online]. Available: `https://unity.com/solutions/automotive-transportation-manufacturing`. (accessed: 27.05.2022).

[33] Unity. 'Unityfilm.' (2022), [Online]. Available: `https://unity.com/solutions/film-animation-cinematics`. (accessed: 27.05.2022).

[34] E. R. T. Technology. 'Vrtk.' (2022), [Online]. Available: `https://www.vrtk.io/`. (accessed: 28.05.2022).

[35] E. R. T. Technology. 'Tillia.' (2022), [Online]. Available: `https://www.vrtk.io/tilia.html`. (accessed: 28.05.2022).

[36] E. R. T. Technology. 'Spatial simulator.' (2022), [Online]. Available: https://github.com/ExtendRealityLtd/Tilia.CameraRigs.SpatialSimulator.Unity. (accessed: 28.05.2022).

[37] T. B. Foundation. 'About.' (2022), [Online]. Available: https://www.blender.org/about/. (accessed: 24.05.2021).

[38] Figma. 'Figma.' (2022), [Online]. Available: https://www.figma.com/about/. (accessed: 28.05.2022).

[39] *Explore oculus quest*. [Online]. Available: https://www.oculus.com/quest/features/.

[40] E. R. T. Technology. 'Tillia interactable.' (2022), [Online]. Available: https://github.com/ExtendRealityLtd/Tilia.Interactions.Interactables.Unity. (accessed: 28.05.2022).

[41] E. R. T. Technology. 'Tillia snapzone.' (2022), [Online]. Available: https://github.com/ExtendRealityLtd/Tilia.Interactions.SnapZone.Unity. (accessed: 03.06.2022).

[42] E. R. T. Technology. 'Zinnia.' (2022), [Online]. Available: https://github.com/ExtendRealityLtd/Zinnia.Unity. (accessed: 03.06.2022).

[43] A. Grabowski and J. Jankowski, 'Virtual reality-based pilot training for underground coal miners,' *Safety Science*, vol. 72, pp. 310–314, 2015, ISSN: 0925-7535. DOI: https://doi.org/10.1016/j.ssci.2014.09.017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925753514002276.

[44] J. Sauro. '10 things to know about the system usability scale (sus).' (), [Online]. Available: https://measuringu.com/10-things-sus/. (accessed: 03.06.2022).

[45] F. Heilemann, G. Zimmermann and P. Münster, 'Accessibility guidelines for vr games - a comparison and synthesis of a comprehensive set,' *Frontiers in Virtual Reality*, vol. 2, 2021, ISSN: 2673-4192. DOI: 10.3389/frvir.2021.697504. [Online]. Available: https://www.frontiersin.org/article/10.3389/frvir.2021.697504.

[46] H. Desurvire and M. Kreminski, 'Are game design and user research guidelines specific to virtual reality effective in creating a more optimal player experience? yes, vr play,' in *Design, User Experience, and Usability: Theory and Practice*, A. Marcus and W. Wang, Eds., Cham: Springer International Publishing, 2018, pp. 40–59.

[47] G. Chanel, C. Rebetez, M. Bétrancourt and T. Pun, 'Boredom, engagement and anxiety as indicators for adaptation to difficulty in games,' in *Proceedings of the 12th International Conference on Entertainment and Media in the Ubiquitous Era*, ser. MindTrek '08, Tampere, Finland: Association for Computing Machinery, 2008, pp. 13–17, ISBN: 9781605581972. DOI: 10.

1145/1457199.1457203. [Online]. Available: https://doi.org/10.1145/1457199.1457203.

Appendix A

Second Iteration Questionnaire

Rapport fra «Spørreundersøkelse april 2022»

Innhentede svar pr. 13. juni 2022 14:25

- Leverte svar: **9**
- Påbegynte svar: **0**
- Antall invitasjoner sendt: **0**

Med fritekstsvar

Bakgrunn

Hei,
Vi er to masterstudenter ved NTNU. Vi jobber med et masteroppgave og prøver å få tilbakemelding og forbedringer på vår jobbsmak app.

Hvor mye erfaring har du med dataspill? *

Svar	Antall	Prosent	
Ingen	0	**0 %**	
Lite	4	**44,4 %**	
Noe	3	**33,3 %**	
Mye	1	**11,1 %**	
Veldig mye	1	**11,1 %**	

Hvor mye erfaring har du med VR/AR ? *

VR står for "Virtual Reality" eller virtuell virkelighet.

AR står for "Augmented Reality" eller utvidet virkelighet.

Svar	Antall	Prosent	
Ingen	3	**33,3 %**	
Lite	4	**44,4 %**	
Noe	1	**11,1 %**	
Mye	1	**11,1 %**	
Veldig mye	0	**0 %**	

Kjønn *

Svar	Antall	Prosent	
Mann	3	**33,3 %**	
Kvinne	6	**66,7 %**	
Annet	0	**0 %**	

Alder *

- 42
- 54
- 28
- 38
- 37
- 36
- 56
- 29
- 58

Konteksten

Jeg har prøvd jobbsmak apper tidligere. *

Svar hvor mange ganger, hvis du har prøvd dem tidligere selv om det er lenge siden.

Svar	Antall	Prosent	
Nei	9	**100 %**	
1 gang	0	**0 %**	
2-5 ganger	0	**0 %**	
over 5 ganger	0	**0 %**	
husker ikke hvor mange ganger	0	**0 %**	

Brukervennlighet

Jeg likte å bruke tømrer appen med VR briller *

Svar	Antall	Prosent	
Svært enig	4	**44,4 %**	
Ganske enig	4	**44,4 %**	
Verken enig eller uenig	1	**11,1 %**	
Litt uenig	0	**0 %**	
Svært uenig	0	**0 %**	

Det var lett å forstå hva jeg skulle gjøre i tømrerappen *

Svar	Antall	Prosent	
Svært enig	4	**44,4 %**	
Ganske enig	5	**55,6 %**	
Verken enig eller uenig	0	**0 %**	
Litt uenig	0	**0 %**	
Svært uenig	0	**0 %**	

Det var lett å forstå oppgavene i tømrer appen *

Svar	Antall	Prosent	
Svært enig	5	**55,6 %**	
Ganske enig	4	**44,4 %**	
Verken enig eller uenig	0	**0 %**	
Litt uenig	0	**0 %**	
Svært uenig	0	**0 %**	

Ting (verktøy, gjenstander osv) i tømrer appen oppfører seg som jeg forventer *

Svar	Antall	Prosent	
Svært enig	0	**0 %**	
Ganske enig	7	**77,8 %**	
Verken enig eller uenig	1	**11,1 %**	
Litt uenig	1	**11,1 %**	
Svært uenig	0	**0 %**	

Det var lett å lese tekst der dette forekom *

Svar	Antall	Prosent	
Svært enig	4	**44,4 %**	
Ganske enig	4	**44,4 %**	
Verken enig eller uenig	0	**0 %**	

Svar	Antall	Prosent	
Litt uenig	1	**11,1 %**	
Svært uenig	0	**0 %**	

Tømrer appen ga meg mer innsikt i hva yrket går ut på *

Svar	Antall	Prosent	
Veldig enig	0	**0 %**	
Enig	6	**66,7 %**	
Verken enig eller uenig	3	**33,3 %**	
Litt uenig	0	**0 %**	
Veldig uenig	0	**0 %**	

System Usability Score
Jeg ville likt å bruke dette systemet ofte *

Svar	Antall	Prosent	
Veldig enig	1	**11,1 %**	
Enig	2	**22,2 %**	
Verken enig eller uenig	5	**55,6 %**	
Litt uenig	1	**11,1 %**	
Veldig uenig	0	**0 %**	

Jeg synes at tømrer appen var unødvendig komplisert *

Svar	Antall	Prosent	
Veldig enig	0	**0 %**	
Enig	0	**0 %**	
Verken enig eller uenig	0	**0 %**	
Litt uenig	3	**33,3 %**	
Veldig uenig	6	**66,7 %**	

Jeg synes at tømrer appen var lett å bruke *

Svar	Antall	Prosent	
Veldig enig	1	**11,1 %**	
Enig	7	**77,8 %**	
Verken enig eller uenig	1	**11,1 %**	
Litt uenig	0	**0 %**	
Veldig uenig	0	**0 %**	

Jeg tror at jeg kommer til å trenge assistanse fra en teknisk person for å bruke tømrer appen *

Svar	Antall	Prosent	
Veldig enig	0	**0 %**	
Enig	1	**11,1 %**	
Verken enig eller uenig	2	**22,2 %**	
Litt uenig	4	**44,4 %**	
Veldig uenig	2	**22,2 %**	

Jeg synes at de forskjellige funksjonene i tømrer appen var bra integrert *

Svar	Antall	Prosent	

Svar	Antall	Prosent
Veldig enig	2	22,2 %
Enig	5	55,6 %
Verken enig eller uenig	1	11,1 %
Litt uenig	1	11,1 %
Veldig uenig	0	0 %

Jeg synes det er for mye inkonsistens i tømrer appen *

Svar	Antall	Prosent
Veldig enig	0	0 %
Enig	0	0 %
Verken enig eller uenig	3	33,3 %
Litt uenig	6	66,7 %
Veldig uenig	0	0 %

Jeg vil se for meg at de fleste folk vil lære seg å bruke tømrer appen veldig fort *

Svar	Antall	Prosent
Veldig enig	2	22,2 %
Enig	7	77,8 %
Verken enig eller uenig	0	0 %
Litt uenig	0	0 %
Veldig uenig	0	0 %

Jeg synes at tømrer appen var veldig tungvint å bruke *

Svar	Antall	Prosent
Veldig enig	0	0 %
Enig	0	0 %
Verken enig eller uenig	0	0 %
Litt uenig	6	66,7 %
Veldig uenig	3	33,3 %

Jeg følte meg veldig trygg på det jeg gjorde i tømrer appen *

Svar	Antall	Prosent
Veldig enig	3	33,3 %
Enig	5	55,6 %
Verken enig eller uenig	0	0 %
Litt uenig	1	11,1 %
Veldig uenig	0	0 %

Jeg trengte å lære masse ting før jeg kunne begynne å bruke tømrer appen *

Svar	Antall	Prosent
Veldig enig	1	11,1 %
Enig	0	0 %
Verken enig eller uenig	2	22,2 %
Litt uenig	0	0 %
Veldig uenig	6	66,7 %

Karriereveiledning

Jeg synes at jobbsmak appene er et godt verktøy i karrieraviledning *

Svar	Antall	Prosent	
Veldig enig	2	**22,2 %**	
Enig	5	**55,6 %**	
Verken enig eller uenig	2	**22,2 %**	
Litt uenig	0	**0 %**	
Veldig uenig	0	**0 %**	

Er det yrker som hadde vært spesielt interessant for deg å prøve ut i jobbsmak, isåfall hvilke? *

- Sveiser?
- Nei
- Service med språklæring
- Nei
- Kokk
- Nei
- Veileder
- Politi
- Bilpleier,sveiser, maler

Slike jobbsmak-apper bør inngå som en del av NAV-tilbud til arbeidssøkere. *

Svar	Antall	Prosent	
Veldig enig	3	**33,3 %**	
Enig	4	**44,4 %**	
Verken enig eller uenig	2	**22,2 %**	
Litt uenig	0	**0 %**	
Veldig uenig	0	**0 %**	

NAV trenger flere jobbsmak-apper til flere yrker. *

Svar	Antall	Prosent	
Veldig enig	3	**33,3 %**	
Enig	3	**33,3 %**	
Verken enig eller uenig	3	**33,3 %**	
Litt uenig	0	**0 %**	
Veldig uenig	0	**0 %**	

Har du noen kommentarer til appen eller forslag til forbedringer?

Takk for din tilbakemelding!!!

Se nylige endringer i Nettskjema

Appendix B

Third Iteration Questionnaire

B.1 End Users

Rapport fra «Spørreundersøkelse 10.5.22 »

Innhentede svar pr. 13. juni 2022 14:27

- Leverte svar: **5**
- Påbegynte svar: **0**
- Antall invitasjoner sendt: **0**

Med fritekstsvar

NTNU
IMTEL – Innovative Immersive Technologies for Learning

nav

Bakgrunn

Hei,
Vi er to masterstudenter ved NTNU. Vi jobber med et masteroppgave og prøver å få tilbakemelding og forbedringer på vår jobbsmak app.

Hvor mye erfaring har du med dataspill? *

Svar	Antall	Prosent	
Ingen	0	**0 %**	
Lite	0	**0 %**	
Noe	2	**40 %**	
Mye	0	**0 %**	
Veldig mye	3	**60 %**	

Hvor mye erfaring har du med VR/AR ? *

VR står for "Virtual Reality" eller virtuell virkelighet.

AR står for "Augmented Reality" eller utvidet virkelighet.

Svar	Antall	Prosent	
Ingen	0	**0 %**	
Lite	2	**40 %**	
Noe	2	**40 %**	
Mye	0	**0 %**	
Veldig mye	1	**20 %**	

Kjønn *

Svar	Antall	Prosent	
Mann	3	**60 %**	
Kvinne	2	**40 %**	
Annet	0	**0 %**	

Alder *

- 22
- 23
- 20
- 23
- 20

Konteksten

Jeg har prøvd jobbsmak apper tidligere. *

Svar hvor mange ganger, hvis du har prøvd dem tidligere selv om det er lenge siden.

Svar	Antall	Prosent	

Svar	Antall	Prosent	
Nei	2	**40 %**	
1 gang	2	**40 %**	
2-5 ganger	0	**0 %**	
over 5 ganger	0	**0 %**	
husker ikke hvor mange ganger	1	**20 %**	

Brukervennlighet

Jeg likte å bruke tømrer appen med VR briller *

Svar	Antall	Prosent	
Veldig enig	1	**20 %**	
Enig	4	**80 %**	
Verken enig eller uenig	0	**0 %**	
Litt uenig	0	**0 %**	
Veldig uenig	0	**0 %**	

Det var lett å forstå hva jeg skulle gjøre i tømrerappen *

Svar	Antall	Prosent	
Veldig enig	2	**40 %**	
Enig	2	**40 %**	
Verken enig eller uenig	1	**20 %**	
Litt uenig	0	**0 %**	
Veldig uenig	0	**0 %**	

Det var lett å forstå oppgavene i tømrer appen *

Svar	Antall	Prosent	
Veldig enig	2	**40 %**	
Enig	2	**40 %**	
Verken enig eller uenig	1	**20 %**	
Litt uenig	0	**0 %**	
Veldig uenig	0	**0 %**	

Opplæringsdelen i starten ga meg god nok kunnskap til å bruke tømrer appen *

Svar	Antall	Prosent	
Veldig enig	2	**40 %**	
Enig	2	**40 %**	
Verken enig eller uenig	1	**20 %**	
Litt uenig	0	**0 %**	
Veldig uenig	0	**0 %**	

Ting (verktøy, gjenstander osv) i tømrer appen oppfører seg som jeg forventer *

Svar	Antall	Prosent	
Veldig enig	2	**40 %**	
Enig	2	**40 %**	
Verken enig eller uenig	1	**20 %**	
Litt uenig	0	**0 %**	

Svar	Antall	Prosent	
Veldig uenig	0	0 %	

Det var lett å lese tekst der dette forekom *

Svar	Antall	Prosent	
Veldig enig	1	20 %	
Enig	2	40 %	
Verken enig eller uenig	1	20 %	
Litt uenig	1	20 %	
Veldig uenig	0	0 %	

Tømrer appen ga meg mer innsikt i hva yrket går ut på *

Svar	Antall	Prosent	
Veldig enig	1	20 %	
Enig	3	60 %	
Verken enig eller uenig	1	20 %	
Litt uenig	0	0 %	
Veldig uenig	0	0 %	

System Usability Score
Jeg ville likt å bruke dette systemet ofte *

Svar	Antall	Prosent	
Veldig enig	2	40 %	
Enig	2	40 %	
Verken enig eller uenig	1	20 %	
Litt uenig	0	0 %	
Veldig uenig	0	0 %	

Jeg synes at tømrer appen var unødvendig komplisert *

Svar	Antall	Prosent	
Veldig enig	1	20 %	
Enig	0	0 %	
Verken enig eller uenig	0	0 %	
Litt uenig	1	20 %	
Veldig uenig	3	60 %	

Jeg synes at tømrer appen var lett å bruke *

Svar	Antall	Prosent	
Veldig enig	4	80 %	
Enig	1	20 %	
Verken enig eller uenig	0	0 %	
Litt uenig	0	0 %	
Veldig uenig	0	0 %	

Jeg tror at jeg kommer til å trenge assistanse fra en teknisk person for å bruke tømrer appen *

Svar	Antall	Prosent	
Veldig enig	1	20 %	

Svar	Antall	Prosent	
Enig	0	0 %	
Verken enig eller uenig	1	20 %	
Litt uenig	1	20 %	
Veldig uenig	2	40 %	

Jeg synes at de forskjellige funksjonene i tømrer appen var bra integrert *

Svar	Antall	Prosent	
Veldig enig	3	60 %	
Enig	2	40 %	
Verken enig eller uenig	0	0 %	
Litt uenig	0	0 %	
Veldig uenig	0	0 %	

Jeg synes det er for mye inkonsistens i tømrer appen *

Svar	Antall	Prosent	
Veldig enig	0	0 %	
Enig	1	20 %	
Verken enig eller uenig	1	20 %	
Litt uenig	1	20 %	
Veldig uenig	2	40 %	

Jeg vil se for meg at de fleste folk vil lære seg å bruke tømrer appen veldig fort *

Svar	Antall	Prosent	
Veldig enig	2	40 %	
Enig	3	60 %	
Verken enig eller uenig	0	0 %	
Litt uenig	0	0 %	
Veldig uenig	0	0 %	

Jeg synes at tømrer appen var veldig tungvint å bruke *

Svar	Antall	Prosent	
Veldig enig	0	0 %	
Enig	0	0 %	
Verken enig eller uenig	2	40 %	
Litt uenig	0	0 %	
Veldig uenig	3	60 %	

Jeg følte meg veldig trygg på det jeg gjorde i tømrer appen *

Svar	Antall	Prosent	
Veldig enig	3	60 %	
Enig	1	20 %	
Verken enig eller uenig	1	20 %	
Litt uenig	0	0 %	
Veldig uenig	0	0 %	

Jeg trengte å lære masse ting før jeg kunne begynne å bruke tømrer appen *

Svar	Antall	Prosent	
Veldig enig	0	**0 %**	
Enig	0	**0 %**	
Verken enig eller uenig	0	**0 %**	
Litt uenig	2	**40 %**	
Veldig uenig	3	**60 %**	

Har du noen kommentarer til appen eller forslag til forbedringer?

- Utvide funskjonalitet av redskap alik at man kan spikre alle planker og ikke bare de som er del av oppgaven
- veldig kult konsept (: liker måten det var lagt opp på og det var enkelt å forstå !!
- Det manglet en forklaring på ene delen av oppgaven, Ellers er alt greit.

Takk for din tilbakemelding!!!

Se nylige endringer i Nettskjema

B.2 Nav Employees

Rapport fra «Spørreundersøkelse april 2022 kopi»

Innhentede svar pr. 13. juni 2022 14:28

- Leverte svar: **5**
- Påbegynte svar: **0**
- Antall invitasjoner sendt: **0**

Med fritekstsvar

NTNU IMTEL – Innovative Immersive Technologies for Learning nav

Bakgrunn

Hei,
Vi er to masterstudenter ved NTNU. Vi jobber med et masteroppgave og prøver å få tilbakemelding og forbedringer på vår jobbsmak app.

Hvor mye erfaring har du med dataspill? *

Svar	Antall	Prosent	
Ingen	1	**20 %**	
Lite	3	**60 %**	
Noe	0	**0 %**	
Mye	1	**20 %**	
Veldig mye	0	**0 %**	

Hvor mye erfaring har du med VR/AR ? *

VR står for "Virtual Reality" eller virtuell virkelighet.

AR står for "Augmented Reality" eller utvidet virkelighet.

Svar	Antall	Prosent	
Ingen	1	**20 %**	
Lite	2	**40 %**	
Noe	2	**40 %**	
Mye	0	**0 %**	
Veldig mye	0	**0 %**	

Kjønn *

Svar	Antall	Prosent	
Mann	3	**60 %**	
Kvinne	2	**40 %**	
Annet	0	**0 %**	

Alder *

- 32
- 44
- 54
- 51
- 53

Konteksten

Jeg har prøvd jobbsmak apper tidligere. *

Svar hvor mange ganger, hvis du har prøvd dem tidligere selv om det er lenge siden.

Svar	Antall	Prosent	

Svar	Antall	Prosent	
Nei	1	20 %	
1 gang	1	20 %	
2-5 ganger	0	0 %	
over 5 ganger	3	60 %	
husker ikke hvor mange ganger	0	0 %	

Brukervennlighet

Jeg likte å bruke tømrer appen med VR briller *

Svar	Antall	Prosent	
Svært enig	4	80 %	
Ganske enig	1	20 %	
Verken enig eller uenig	0	0 %	
Litt uenig	0	0 %	
Svært uenig	0	0 %	

Det var lett å forstå hva jeg skulle gjøre i tømrerappen *

Svar	Antall	Prosent	
Svært enig	3	60 %	
Ganske enig	2	40 %	
Verken enig eller uenig	0	0 %	
Litt uenig	0	0 %	
Svært uenig	0	0 %	

Det var lett å forstå oppgavene i tømrer appen *

Svar	Antall	Prosent	
Svært enig	3	60 %	
Ganske enig	2	40 %	
Verken enig eller uenig	0	0 %	
Litt uenig	0	0 %	
Svært uenig	0	0 %	

Ting (verktøy, gjenstander osv) i tømrer appen oppfører seg som jeg forventer *

Svar	Antall	Prosent	
Svært enig	3	60 %	
Ganske enig	2	40 %	
Verken enig eller uenig	0	0 %	
Litt uenig	0	0 %	
Svært uenig	0	0 %	

Det var lett å lese tekst der dette forekom *

Svar	Antall	Prosent	
Svært enig	4	80 %	
Ganske enig	1	20 %	
Verken enig eller uenig	0	0 %	
Litt uenig	0	0 %	

Svar	Antall	Prosent
Svært uenig	0	**0 %**

Tømrer appen ga meg mer innsikt i hva yrket går ut på *

Svar	Antall	Prosent
Veldig enig	1	**20 %**
Enig	4	**80 %**
Verken enig eller uenig	0	**0 %**
Litt uenig	0	**0 %**
Veldig uenig	0	**0 %**

System Usability Score
Jeg ville likt å bruke dette systemet ofte *

Svar	Antall	Prosent
Veldig enig	2	**40 %**
Enig	3	**60 %**
Verken enig eller uenig	0	**0 %**
Litt uenig	0	**0 %**
Veldig uenig	0	**0 %**

Jeg synes at tømrer appen var unødvendig komplisert *

Svar	Antall	Prosent
Veldig enig	0	**0 %**
Enig	0	**0 %**
Verken enig eller uenig	0	**0 %**
Litt uenig	1	**20 %**
Veldig uenig	4	**80 %**

Jeg synes at tømrer appen var lett å bruke *

Svar	Antall	Prosent
Veldig enig	3	**60 %**
Enig	2	**40 %**
Verken enig eller uenig	0	**0 %**
Litt uenig	0	**0 %**
Veldig uenig	0	**0 %**

Jeg tror at jeg kommer til å trenge assistanse fra en teknisk person for å bruke tømrer appen *

Svar	Antall	Prosent
Veldig enig	0	**0 %**
Enig	1	**20 %**
Verken enig eller uenig	0	**0 %**
Litt uenig	1	**20 %**
Veldig uenig	3	**60 %**

Jeg synes at de forskjellige funksjonene i tømrer appen var bra integrert *

Svar	Antall	Prosent
Veldig enig	2	**40 %**

Svar	Antall	Prosent	
Enig	3	**60 %**	
Verken enig eller uenig	0	**0 %**	
Litt uenig	0	**0 %**	
Veldig uenig	0	**0 %**	

Jeg synes det er for mye inkonsistens i tømrer appen *

Svar	Antall	Prosent	
Veldig enig	0	**0 %**	
Enig	0	**0 %**	
Verken enig eller uenig	0	**0 %**	
Litt uenig	2	**40 %**	
Veldig uenig	3	**60 %**	

Jeg vil se for meg at de fleste folk vil lære seg å bruke tømrer appen veldig fort *

Svar	Antall	Prosent	
Veldig enig	2	**40 %**	
Enig	3	**60 %**	
Verken enig eller uenig	0	**0 %**	
Litt uenig	0	**0 %**	
Veldig uenig	0	**0 %**	

Jeg synes at tømrer appen var veldig tungvint å bruke *

Svar	Antall	Prosent	
Veldig enig	0	**0 %**	
Enig	0	**0 %**	
Verken enig eller uenig	0	**0 %**	
Litt uenig	1	**20 %**	
Veldig uenig	4	**80 %**	

Jeg følte meg veldig trygg på det jeg gjorde i tømrer appen *

Svar	Antall	Prosent	
Veldig enig	1	**20 %**	
Enig	3	**60 %**	
Verken enig eller uenig	0	**0 %**	
Litt uenig	1	**20 %**	
Veldig uenig	0	**0 %**	

Jeg trengte å lære masse ting før jeg kunne begynne å bruke tømrer appen *

Svar	Antall	Prosent	
Veldig enig	0	**0 %**	
Enig	0	**0 %**	
Verken enig eller uenig	0	**0 %**	
Litt uenig	3	**60 %**	
Veldig uenig	2	**40 %**	

Karriereveiledning

Jeg synes at jobbsmak appene er et godt verktøy i karrieraviledning *

Svar	Antall	Prosent	
Veldig enig	2	**40 %**	
Enig	3	**60 %**	
Verken enig eller uenig	0	**0 %**	
Litt uenig	0	**0 %**	
Veldig uenig	0	**0 %**	

Er det yrker som hadde vært spesielt interessant for deg å prøve ut i jobbsmak, isåfall hvilke? *

- Dagligvare, budbilsjåfør/ sjåfør, lastebilsjåfør
- Yrker som ikke et vanlige
- Butikk, helsefag, barnehage
- Mekanikker
- Tømrer

Slike jobbsmak-apper bør inngå som en del av NAV-tilbud til arbeidssøkere. *

Svar	Antall	Prosent	
Veldig enig	0	**0 %**	
Enig	4	**80 %**	
Verken enig eller uenig	1	**20 %**	
Litt uenig	0	**0 %**	
Veldig uenig	0	**0 %**	

NAV trenger flere jobbsmak-apper til flere yrker. *

Svar	Antall	Prosent	
Veldig enig	0	**0 %**	
Enig	5	**100 %**	
Verken enig eller uenig	0	**0 %**	
Litt uenig	0	**0 %**	
Veldig uenig	0	**0 %**	

Har du noen kommentarer til appen eller forslag til forbedringer?

- Tømrer: liten beskrivelse av hva oppdraget er, vi bygger et hus og du skal lage ramme til dør og legge kledning?

Takk for din tilbakemelding!!!

Se nylige endringer i Nettskjema

A. Johansen, E. Johansen

CarpenterVR

NTNU
Norwegian University of
Science and Technology