Evaluating Virtual Reality as a Medium for Usability Testing on Inflight Entertainment Applications

Fredrik Voigt and Rasmus Andersson

DEPARTMENT OF DESIGN SCIENCES
FACULTY OF ENGINEERING LTH | LUND UNIVERSITY
2023

MASTER THESIS

TACTEL

Evaluating Virtual Reality as a Medium for Usability Testing on Inflight Entertainment Applications

Fredrik Voigt
fr2405vo-s@student.lu.se

Rasmus Andersson
ra7232an-s@student.lu.se

12 juni 2023

Master's thesis work carried out at Tactel AB.

Supervisors: Joakim Eriksson, joakim.eriksson@design.lth.se
Joel Jonsson, joel.jonsson@tactel.se

Examiner: Günter Alce, gunter.alce@design.lth.se

Evaluating Virtual Reality as a Medium for Usability Testing on Inflight Entertainment Applications

Published by
Department of Design Sciences
Faculty of Engineering LTH, Lund University
P.O Box 118, SE-221 00 Lund, Sweden

Subject: Interaction Design (MAMM01),
Division: Ergonomics and Aerosol Technology
Supervisor: Joakim Eriksson
Examiner: Günter Alce
Company supervisor: Joel Jonsson

Abstract

This thesis aims to evaluate the feasibility of using virtual reality as a medium for usability testing. Usability testing has long been an important part of any iterative development process. Most usability testing is, however, done in a lab setting, outside the product's natural environment. Virtual Reality (VR) on the other hand can engross a user in any environment.

This thesis was done in an iterative manner, splitting it into two iterations both with a development phase and a testing phase. The first iteration's development phase involved creating the VR application which was going to simulate an aeroplane environment for the participant. The application created a one-to-one mapping between a physical screen that the users tapped on and its streamed counterpart which they saw in VR. The testing phase then continued by writing a full test plan and conducting a usability test on an In-Fligt Entertainment (IFE) map application. The second iteration took the results and feedback from the first and improved on the most lacking parts, namely the mapping between the virtual and real-life screens. It contained the same steps as the first iteration but improved only on existing aspects rather than creating anything new.

The final VR application put forth after the second iteration showed that VR can be used as a medium for usability testing. Using VR does increase costs and workload for the test researchers but the negative aspects are outweighed by the benefits. The participants' answers show that VR makes the tests more interesting and more realistic. For some users, the fact that the moderator is hidden also makes them more confident and want to explore the application further.

Keywords: virtual reality, VR, usability evaluation, usability testing, UX

Sammanfattning

Syftet med denna rapport är att utvärdera möjligheten att använda Virtuell Verklighet (VR) som ett medium för användbarhetstestning. Användbarhetstestning är en viktig del av alla iterativa utvecklingsprocesser. Vanligtvis utförs testerna i en laboratoriemiljö utanför produkternas naturliga miljö. Genom att använda VR kan dock användare uppleva testmiljön på ett mer realistiskt sätt.

Studien genomfördes på ett iterativt sätt och delades upp i två iterationer, var och en med en utvecklingsfas och en testfas. Under den första utvecklingsfasen skapades en VR-applikation som simulerade en flygplansmiljö för deltagarna. Applikationen skapade en en-till-en-koppling mellan en fysisk skärm som användarna interagerade med och dess virtuella motsvarighet. I testfasen genomfördes ett användbarhetstest av en kartapplikation för ett flygunderhållningsystem med en fullständig testplan.

Den andra iterationen byggde på resultaten och feedbacken från den första iterationen och fokuserade på att förbättra kopplingen mellan de virtuella och verkliga skärmarna. Iterationen inkluderade samma steg som den första iterationen, men fokuserade på att förbättra befintliga aspekter istället för att skapa något nytt.

Den slutliga VR-applikationen visade att virtuell verklighet kan användas som ett medium för användbarhetstestning. Även om användning av VR kan öka utvecklingskostnaden for tester och arbetsbelastningen för testforskarna, så väger fördelarna upp för de negativa aspekterna. Deltagarna ansåg att VR gjorde testerna mer intressanta och realistiska. För vissa användare gjorde det faktum att moderatorn var dold också att de kände sig mer självsäkra och villiga att utforska applikationen ytterligare.

Nyckelord: virtuell verklighet, VR, användbarhetsutvärdering, användbarhetstest, UX

Acknowledgements

We would like to thank our supervisors Joakim Eriksson and Joel Jonsson for their continuous support during this thesis. Also a big thank you to the design team at Tactel for valuable discussions and inputs. Thank you to everyone who participated in the study and provided valuable data for our thesis. Finally, we would like to thank Tactel and all employees for taking us in and making us feel like we were part of the family.

Acronyms and Abbreviations

- **AR** - Augmented Reality
- **HMD** - Head Mounted Display
- **IFE** - Inflight Entertainment
- **MR** - Mixed Reality
- **OBS** - Open Broadcast Software
- **PQ** - Presence Questionnaire
- **RL** - Real Life
- **RQ** - Research Question
- **VR** - Virtual Reality

Contents

Chapter 1
Introduction

This chapter introduces the thesis. It will present the thesis's background, purpose, goals, research questions, scope, limitations, and related work. It will also introduce Tactel, the company that this thesis has been done in collaboration with.

1.1 Background

Usability testing has long been an essential part of the iterative development processes [1]. It falls under the evaluation part of the iterative design life cycle which, as can bee seen in figure 1.1, is one of the four main parts of cycle. These tests aim to find out which aspects of a prototype or product work, which need adjustment or fixing, and what the prototype or product is missing. Thereafter, the necessary changes can be implemented and then a new usability test can be performed. Usually, this process will be iterated multiple times before a product is considered complete. Often the product will never be seen as perfect and this process continues after the initial release of the product. However, due to their importance and the need for multiple iterations of a product, usability tests are often done in lab settings. Those settings are usually cheaper and more controlled than natural settings, but they do introduce negative aspects such as the effects of being observed [2]. It is stated that people perform differently, especially when faced with more difficult tasks when they are observed. Additionally, sometimes testing a product in its intended environment might yield different results than in a lab simply because of different, more realistic distractions and a more natural setting. Unfortunately, testing outside of a lab can often be more costly and gives less control of the experiment to the developers [2].

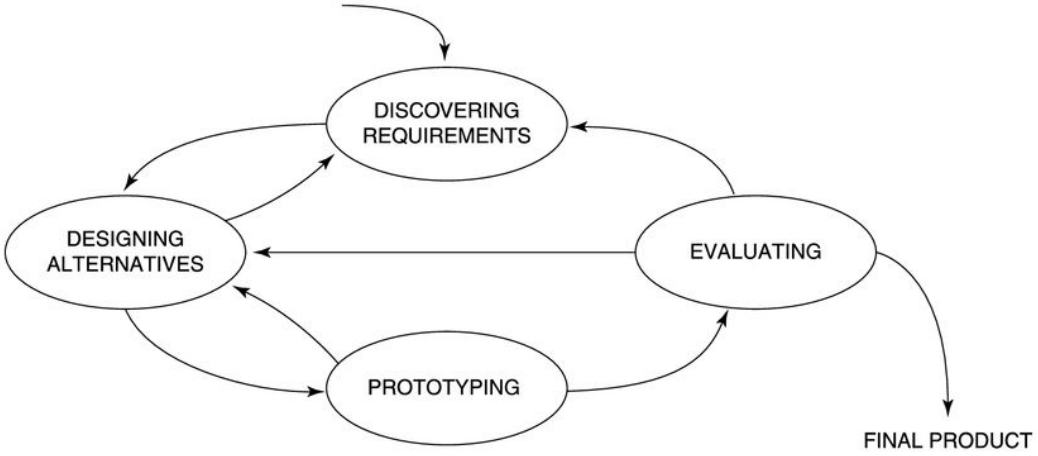

Figure 1.1: A simple interaction design life cycle model [1].

Virtual Reality (VR) is a concept many people at least have heard about since its use and popularity has increased significantly the last couple of years [3]. VR allows the users to digitally travel to another place, replacing the room they were in with a whole new environment without actually moving the user anywhere. It then only seems natural to consider if VR can be used to improve usability testing. Allowing tests to be conducted in a more natural setting while also keeping the researchers in control. Additionally, VR could remove the observers and moderator from the view of the participants and thus the effects of being observed. However, since VR does not create things in actuality its effects on usability testing are difficult to predict and ultimately unknown.

Tactel

This thesis is done in collaboration with Tactel AB. Tactel, founded in 1995, is a digital interaction agency and became a part of Panasonic Avionics Corporation in 2015. Today the company has around 100 employees in Sweden [4]. One of their main projects is designing and developing their In-flight Entertainment (IFE) map application, Arc, for Panasonic and their customers. Arc, together with multiple other projects creates a work environment that contains a large amount of both internal testing and usability testing. This is why a collaboration between us and Tactel worked so well, we needed a stable testing ground and they were interested in exploring a possible improvement in their existing testing environment.

1.2 Purpose and Goals

This thesis aims to investigate if Virtual Reality can be used in usability testing of IFE applications. Can tests be improved by immersing users in a more correct environment and still provide accurate results?

Goals

- Develop a Virtual Reality testbed in order to evaluate the usability of a IFE product.
- Evaluate the feasibility of using Virtual Reality as a medium for usability testing.

1.3 Research questions

What are the limitations and drawbacks of the current method for usability testing when a natural setting is not feasible?

What is gained and what is lost when using Virtual Reality as a medium for usability testing compared to normal usability testing?

How does Virtual Reality affect usability testing when natural settings otherwise cannot be reconstructed in regards to, cost, efficiency, perceived product experience, subjective workload, and validity?

1.4 Scope and Limitations

Due to the time restrictions of our thesis work, a scope and some limitations was needed to manage our time.

The scope of this thesis is to focus on and limit our testing to only improving the usability tests. Meaning the study is only interested in the difference between the results of the usability tests and not the results themselves, i.e. we will not analyse the usability of an application as a product. This thesis is also limited to only comparing our methods with already existing in-office/lab testing. That means we will not compare the results with tests made in the product's natural environment i.e. tests made on a flying plane as that scale is beyond our scope's possibilities.

The main limiting factor for us is hardware. We will only develop for and test our theories on the Meta quest pro headset as well as an already released version of an IFE application made by Tactel, more specifically Arc. Further limitation comes in the form of participants. Tactel usually mainly tests on their employees, using those who work on different projects than the one tested. This is both due to time factors but also administrative costs of having non-employees enter the office. The second factor will be the biggest limiter for us, thus limiting our test group to Tactel employees and a few close friends and relatives.

1.5 Global Goals

In 2015 the United Nations set out to create a set of goals to achieve by 2030 [5]. In total there are 17, which make up the Sustainable Development Goals. They range from ending poverty and hunger to combating climate change and promote sustainable growth.

This thesis is closest connected to two of the 17 goals. It is directly connected to goal nine, *Industry, Innovation and Infrastructure* (logo displayed in figure 1.2) [6], as we aspire to promote innovation through improved usability testing. It is also indirectly connected to goal 13, *Climate Action* (logo displayed in figure 1.3) [7]. Testing an IFE application currently is rarely, if ever, done on an actual aeroplane due to costs and time. Additionally, using an actual aeroplane as the environment would also impact the climate, especially if the testing should be done while airborne. Testing in VR on the other hand opens up the possibility of testing in an aeroplane environment without negatively impacting the climate.

Figure 1.2: UN Goal 9, *Industry, Innovation and Infrastructure* [6].

Figure 1.3: UN Goal 13, *Climate Action* [7].

1.6 Related work

Previous research has been made on the effect of different environments when conducting different user-based tests. As an example, Sauer et al. [2] examined remote and classical field testing as alternatives to lab-based usability testing. They give an interesting insight into the difference of being in a more natural setting when participants test a product [2]. This peaked our interest as we saw the possibility of enhancing their findings with our main interest, VR development. VR has in recent years increased rapidly in education, healthcare, entertainment, and many other areas. In 2020, the VR market was estimated to have reached $6.1 billion and is estimated to reach $20.9 billion by 2025 [3]. As VR becomes more widely used we believe it to be the next logical step to use as a medium for testing and research. Which some already has seen and begun testing. Freitas et al. [8] performed a literature review regarding VR on product usability testing, highlighting multiple other papers researching

this topic and showing the rise in interest for this use case [8]. Others have already begun using it in similar ways. Kinateder et al. [9] studied using VR as a medium for testing fire safety scenarios where it was too expensive or dangerous to reconstruct it in real life. They found that even though it does not invoke the exact same responses it performs well enough to be considered useful [9]. Picka et al. [10] have also studied using VR as a way to simulate situations that are difficult to imitate in reality without risking the health of users in domains of product design, like medical device design [10].

1.7 Distribution of work

Most of the work in this project has been done by us together. Designing the application, reading literature, hunting for bugs in the code, discussing and analysing data and so on have been together and neither could say that the other has done more or less. There are of course a few aspects of the project which have demanded more time and focus in the same time frame which meant some distribution was required. Below is a list for each person with the few bigger parts of the project which they have been in charge of.

Fredrik

- Anchor placement and screen alignment in VR application
- Setup menu in VR application
- Presence questionnaire and analysis of it
- Moderator during usability tests

Rasmus

- Simulation environment (Aeroplane, passengers, sound, clouds, etc.)
- Test plan
- Observer during usability tests

Chapter 2
Theory and Technology

In this chapter the essential theory and related technologies are presented. Firstly, Usability Evaluation will be introduced together with the concepts from it which are used in this study. Followed by defining what Virtual Reality is and its most important concepts. Lastly, it describes some additional programs and software used.

2.1 Usability Evaluation

Usability evaluation, composed of qualitative and quantitative research, is the process of assessing the user-friendliness of a system or product and whether or not it satisfies users' needs. There are multiple ways to conduct such evaluations. These evaluation methods can be classified into three broad categories, depending on the setting, user involvement, and level of control [1].

- **Controlled settings directly involving users:** Users' activities are controlled and moderated in a setting where they can be observed and recorded, to determine how the user group interacts with the product and where potential problems lie. The main methods are usability testing and experiments.
- **Natural settings involving users:** Consists of little to no involvement from the testers and instead lets users explore the product in the real world to determine how well it works. Here field studies are the main testing method.
- **Any settings not directly involving users:** Consultants, experts, and researchers study, critique, and model aspects of the interface/product to identify the most obvious usability problems. Here there are multiple different methods including inspections, heuristics, walkthroughs, models, and analytics.

There are pros and cons for each evaluation category. For example, lab-based studies like usability testing are good at revealing usability problems, but they are poor at capturing the context of use [1]. For this thesis our main focus lies specifically on usability testing and how a connection between the controlled and the natural setting can be created. Usability testing is usually one of the main parts of any iterative design process and is often better done excessively rather than too little. However, in order to fully define usability testing a few other terms are first needed to be defined.

Usability

To understand what actually is tested when doing usability testing we first have to determine what is meant by *usability*. To begin we look at the ISO definition. ISO 9241-11 states:

Extent to which a system, product or service can be used by specified users to achieve specified goals with effectiveness, efficiency, and satisfaction in a specified context of use [11].

What counts as "effectiveness, efficiency and satisfaction" is highly subjective and in many ways difficult to specify when determining usability through testing. A user could be satisfied without reaching the speed in use that the developer had planned when designing the product. In their book Rubin and Chisnell concretise usability as *" when a product or service is truly usable the user can do what he or she wants to do the way he or she expects to be able to do it, without hindrance, hesitation, or questions"* [12]. Taken from these two definitions is that it is near impossible to define usability as one thing that is universal for every product. The main thing is that when designing a product the focus lies on that the end user can use the product without having to struggle with it.

User experience

Another aspect tested during usability testing is the User Experience (UX) of the product. The ISO definition of UX is:

User's perceptions and responses that result from the use and/or anticipated use of a system, product or service [11]

Which means that UX is all about how a product feels to use and interact with [1]. Furthermore, an individual's *experience* with a product is subjective but UX is not. The principles of UX instead look more at the collective experience of multiple users, making it more objective. It is also important to point out that UX is not something that can be designed but rather should be designed for. UX is all about the emotional response of the users using a product and thus depends on multiple factors not necessarily linked to the product itself, such as previous knowledge and environment [1].

What is Usability testing?

The term usability testing refers to a process that employs people as testing participants who act as representatives of the target audience to evaluate the degree to which a product meets specific usability criteria. The tests can be versatile ranging from true classical experiments with large sample sizes and complex test designs to very informal qualitative studies with only a single participant. Each testing approach has different objectives, as well as different time and resource requirements [12]. This thesis will keep to the more true classical experiment by setting up a series of tasks that every participant will perform as well as keeping our interaction with the participants consistent and formal. The main difference being that our study will be scaled accordingly to be able to be performed with a limited number of researchers.

2.1.1 Test plan

The test plan is a core part of usability testing as it acts as the blueprint of the whole process. By creating a plan one creates a solid groundwork that makes sure that all tests conducted are carried out in the same way. By ensuring consistency the risk of tainting the result from errors made by us and not the participants, is minimised. Another important reason in favour of creating a test plan is to ensure total clarity within the testing team. By having everything written in black and white each person knows what is needed to conduct a test session and what equipment is needed and what each person should do. It also makes it easier to bring in new people or explain the tests to an executive or other external personnel if needed [12]. A typical test plan contains the following points:

- **Purpose and goals** - A short description of why the test is conducted.
- **Research questions** - The questions the study aims to answer.
- **Participant characteristics** - Description of who the target audience is and thus the requirements for participants.
- **Method** - How each step of the test will be conducted and what is needed for it in terms of materials and personnel.
- **Task list** - Scenario description, list of tasks and subtasks to be completed by the participant as well as criteria for success and maximum time for each task.
- **Test environment, equipment and logistics** - How the lab/test environment should be set up and what equipment the test needs.
- **Roles** - Description and division of each role in the test group. For example, test moderator, observer, technician, timekeeper, etc.
- **Data to be collected and evaluation measures** - Details what data each task aims to collect to answer the research questions.
- **Presentation of results** - How the results will be processed and presented to other parties like dev-teams, other research teams, or executives.

2.1.2 Data gathering

To gather people's opinions and results, ways to collect and categorise that data are needed. Described below are some key words for discussing the topic as well as some of the most common methods for gathering data.

Quantitative data

Quantitative data refers to data that either is in the form of numbers or easily can be translated to numbers [1]. This type of data often gives a convincing argument and can lay the foundation for most decisions. Additionally, quantitative data is great for finding things such as magnitudes and amounts within a population. Unfortunately, it is also relatively easy to manipulate quantitative data, for example, one could perform a study on a small population and then present percentages instead of fractions [1]. Therefore, it is important to always keep clarity with how and why data is presented.

Qualitative data

Qualitative data covers more complex data, that is, words, pictures, descriptions, etc. [1]. It is thus often more detailed than quantitative data but also usually more open and varied. Analysing qualitative data often focuses more on finding patterns and themes, which can lead to a greater knowledge of what the users find easy and difficult with a product. Furthermore, qualitative data can also help answer why and how different aspects affect and work within and around a product.

Questionnaires

A great way to quickly collect large amounts of data is with the help of questionnaires. Questionnaires can collect both quantitative and qualitative data but usually focuses mainly on quantitative, as it is easier to answer questions with numbers and scales rather than writing free-form text to answer more nuanced qualitative questions. Its strength lies in how inexpensive of a method it is to gather lots of data as you can easily use online surveys or send questionnaires through mail, reaching hundreds in seconds [1]. Important to note however is that once a questionnaire has been sent out then there is no possibility to change or explain questions. This means that any problems participants encounter with the survey cannot be ratified and might lead to skewed or faulty results. For that reason, it is very important to be thorough when writing questionnaires and pilot test them before sending them out to find the problems in time. With questionnaires there is also the risk of applying one's own bias to the questions, colouring the answer of the participants if you are not careful. To avoid this one can either make sure all questions are as objectively formulated as possible or balance the questions by asking the same question twice, once formulated positively and once negatively.

Interviews

As with questionnaires, interviews can be made to collect both types of data but instead excel at gathering qualitative information. Interviews are great for complementing participants' questionnaire answers by going into more detail on aspects that are hard to quantify as well as opening up the possibility to ask follow-up questions when either party does not understand the other. An interview's greatest strength is the flexibility that comes with being able to continue the conversation. The weakness being of course that each interview takes time and needs a researcher to conduct the whole thing.

There are four main types of interviews: open-ended or unstructured, structured, semi-structured, and group interviews [1]. The first three are most useful for us as one on one interviews. The name of the type signifies where it lands on a scale between the moderator having a conversation with the participant and just asking a set amount of questions and nothing more. The semi-structured is closest to how we have described interviews up until now, a mix of interview types where you use a predetermined set of questions as the basis for the interview but the moderator has free reigns to ask follow-up questions and dig deeper into parts where they feel more information is needed.

During debriefing the moderator can also make use of certain more advanced debriefing techniques. One of these is closely connected to the semi-structured interview, namely the "Devil's advocate" technique [12]. In this technique the test moderator drops their neutral demeanour and instead takes the opposite opinion to the participant's, to ascertain the participant's true feelings about the product. This is an especially useful technique when you suspect that the participants either are reluctant to criticise the product in front of the test researchers or if they are positively or negatively biased towards the product already before the test. The technique, however, is considered advanced as the moderator could easily run the risk of manipulating the answers of the participants instead of getting more true answers if they do not use the method with care. Especially when talking to more vulnerable participant groups like children or the elderly.

2.1.3 Norman's design principles

When talking about interaction design and usability it is mandatory to mention Donald Norman's design principles [13]. From years of experience and research, Norman has distilled much of the core of interaction design into five fundamental design principles. These are presented in the list below and are used either consciously or unconsciously by any developer who creates any product to be interacted with today, including us in this thesis [13].

- **Affordances** - The relationships between how an object looks and what it does. Good affordance means a user should instantly know what they can do with a given object.
- **Signifiers** - Ways of communicating clear and understandable information to the user in the form of icons, labels, words, etc. to help them understand what they can do with the object.
- **Constraints** - Limit the number of possible interactions to steer the user in the direction of using the object as intended. For example, an online form that does not allow users to enter letters into a phone number field.
- **Mappings** - The relationship between control and effect. The idea is that with good design, the controls to something will closely resemble what they affect.
- **Feedback** - Give the user immediate, informative, and appropriate information when they have performed an action that lets them know whether or not it was successful. For example, changing the icon on the tab to a spinner to indicate that a webpage is loading.

2.2 Virtual Reality

Virtual Reality (VR) is a way to engross a user in a digital world. Heim [14] defines VR as a technology that could convince the user that they actually are in a different place. The definition further defines three I:s of VR; immersion, interactivity, and information intensity. In this sense, immersion refers to a device that could isolate a person's senses sufficiently to make them feel transported. Interactivity would be that the device keeps up with the person's ability to move and change position. Finally, information intensity refers to the possibility of the virtual world inducing presence, the feeling of being there [14]. This definition does make VR seem focused on the technology and also does not fully differentiate VR from other types of reality, such as Augmented Reality (AR) or Augmented Virtuality (AV).

In contrast to Heim's definition, VR is often only seen as a part of what as a whole usually is called the Reality-Virtuality (RV) Continuum (figure 2.1). Milgram et al. [15] define a more commonly used definition, namely the RV continuum. In this definition VR is the furthest step from reality since in VR everything is digitally constructed and nothing of the real world is visible. Anything in between is then seen as Mixed Reality (MR) since both real-world objects and virtual objects can be seen and/or interacted with [15].

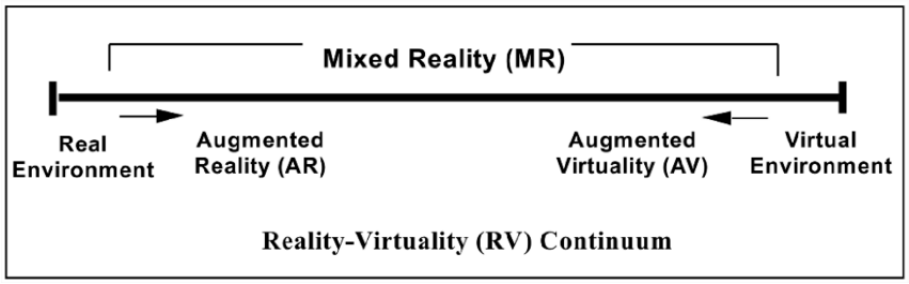

Figure 2.1: The Reality-Virtuality (RV) continuum [15].

Furthermore, Rauschnabe et al. [16] add to the definition of VR by stating that VR can be looked at on a presence (defined below) scale, from atomistic to holistic VR [16]. Atomistic VR refers to VR where the experience mainly is linked to the completion of a certain task, such as viewing blueprints for construction. While holistic VR refers to when the experience is indistinguishable from the real world, that is, when the user can feel more present in the virtual world rather than the real.

Augmented Reality

Augmented Reality is a mixture of reality and virtuality. As can be seen in the RV continuum in figure 2.1, any mixture of reality and virtuality would be considered MR, and AR is the mixture closest to reality. AR takes the real world as its base and then adds virtual objects, such as images, objects, audio, etc, to it [17].

2.2.1 Presence and Immersion

A big part of VR is presence and immersion. Through the years the exact meaning of these terms has been disputed. We have chosen to use the definitions first put forward by Slater [18]. He defines presence as the subjective feeling of "being there" in the virtual world. Furthermore, if measured, presence also includes the extent to which the virtual world becomes the dominant reality and after a virtual experience how much one remembers visiting a place rather than simply remembering using a device or seeing images. On the other hand, Slater defines immersion as a solely objective measure of a system's performance and specifications. That is, for example, a device with a greater field of view, higher resolution, and more speakers for directional sound is more immersive than a device with worse characteristics.

One might think that the visual aspects of a virtual experience have the greatest impact on presence but audio and other non-visual cues are shown to be as important. Potter et al. [19] performed a study testing the effects of audio quality versus the effects of visual quality on presence (by them referred to as immersion). They showed that both had a significant impact on presence and that neither should be prioritised over the other. More specifically, they found that adding reverberation to head-tracked binaural audio produced the same presence as increasing the resolution five folds from 0.5 megapixels to 2.5 [19].

Measuring Presence

In order to evaluate how well a VR application immerses its users in the virtual world one needs to measure presence. Unfortunately, since presence is a subjective matter there is no perfect way to do so. One popular method is to use presence questionnaires. There exist a few such questionnaires, each with different pros and cons. One of them is the *Presence Questionnaire* originally made by Witmer and Singer [20]. The latest found version consists of 32 questions, each scored on a scale of one to seven [21]. Even though their *Presence Questionnaire* is one of the more popular presence questionnaires it is not without its flaws. Slater points out that the questionnaire not always perfectly distinguishes between what is caused by the VR experience and what is caused by the participants' previous experiences [18]. For example, one of the questions in the questionnaire is *How much were you able to control events?* If the VR experience being tested involved say, playing tennis, then a user could interpret the question as "How well did you play tennis?" while the question really tries to find out how much **VR** affected their ability to play tennis. The questions do not differentiate between what is presence-inducing and what is the effect of other factors. As such a user who is bad at tennis might answer the question with a low score even if it was very similar to their real-life experiences and potentially highly presence-inducing. Furthermore, Slater also states that questionnaires in general cannot be used on their own to assess presence. Therefore, using the *Presence Questionnaire* on its own in a vacuum might be a bad idea. However, it can still be useful in combination with other ways to measure. Two of these ways are with proper screening of the participants and a complementing interview regarding their experience [22].

2.2.2 Head Mounted Display

One of the most common ways to experience VR is through the use of a Head Mounted Display (HMD). An HMD is a device that, as one can expect, is worn on the head. Today it usually comes with a high-resolution display, sound, and rotational and translational tracking in all directions. They use two different displays, one for each eye, allowing the display of stereoscopic images, i.e. slightly different images for each eye in order to simulate depth and distance [23]. Another important part of HMDs is the way they let the user interact with the virtual world and its contents. The current standard is to use two hand controllers, however, some HMDs including the one used in this thesis, can use hand tracking [24], allowing the user's own hands to act as controllers. Which can give a more realistic feeling for certain VR experiences. Examples of how it looks in VR when using controllers and hand tracking as controlling scheme can be seen in figures 2.2 and 2.3 respectively.

Figure 2.2: Using controllers as controlling scheme with real hand and controller (left) and virtual controller (right).

Figure 2.3: Using hand tracking as controlling scheme with real hand (left) and virtual hand (right).

Degrees of Freedom

Degrees of Freedom (DoF) refers to the number of basic ways a body can be moved [25]. In regards to HMDs, this also refers to the number of basic movements an HMD can track. In total, there are six DoFs, three translational (up/down, right/left, and forward/backward) and three rotational, one around each axis x, y, and z. The HMDs of today either come with three Dof (3DoF) tracking, only tracking rotational movements, or six DoF (6DoF) tracking, tracking all movements.

2.2.3 Eye tracking

Eye tracking is essentially what it sounds like. It is the process of continuously following the eyes' position and movements. The best outcome is if the process is done purely in a non-invasive manner. Generally, there are three types of measurement devices, i.e. eye trackers. Video (or photo or laser) based eye tracker, electro-oculography, and scleral search coils (or contact lenses) [26]. Most modern eye trackers are video-based devices as they are the most non-invasive and simple to use [27]. They also provide more of a sense of what an individual is looking at rather than just detecting their eye movement. The eye-tracking built into the Meta quest pro used in this thesis is for example such a video-based device [28]. Video-based eye trackers use a source of infrared light, invisible to the human eye, that illuminates the pupil and reflection that is generated on the cornea to track the eye position. Specifically, an infrared camera records the pupil position and corneal reflection, with the eye position provided in raw data as x and y coordinates contained within the field of the camera [26, 27]. The raw data can then be processed using other external software which outputs it to something more readable, e.g. heatmaps.

Eye movements provide valuable insight into various aspects of cognitive processing, making them a useful tool in understanding how people interact with technology and products. Eye tracking has been utilised in human-computer interaction and user experience research for many years, providing additional information for researchers to deduce the underlying meaning of users' actions when engaging with applications and devices [26]. The area a person is looking directly at is usually where they focus their attention. This information can be used in two ways, in the moment it can be seen as an indication of what the person is

thinking about right there and then. Afterwards, heatmaps and other processing tools can then show which parts of the product got the person's attention the most. UX researchers can examine this data and use it to determine a person's level of attention to certain aspects and areas. Which in turn can be used to objectively determine the best placement of specific elements within an interface or product to optimise user experience.

2.2.4 Motion and Cybersickness

Motion sickness is commonly occurring and is caused by a mismatch between sensor modalities [29]. For example, when reading a book on a train, the inner ear senses movement while the eyes are looking at something still. Motion sickness can come suddenly and can lead to symptoms such as dizziness, cold sweats, nausea, and/or headache.

Cybersickness is similar to motion sickness but can, amongst other places, be experienced while in a VR or MR experience [30]. Furthermore, even though the two share most of their symptoms, cybersickness more frequently induces symptoms linked to disorientation, such as dizziness, while motion sickness more frequently induces nausea. The cause of cybersickness is not fully determined though the most common theory, like motion sickness, is sensory mismatch.

2.3 Programs and Software

2.3.1 Unity

In order to develop VR applications it is beneficial to use a game engine, for example Unity. Unity offers easy development of 3D applications on multiple platforms, including both Windows and Android [31]. It also contains a library of VR and AR tool kits [32, 33] to simplify and speed up development, letting users skip reinventing the wheel.

Unity shader graph

Unity shader graph [34] is a tool that simplifies the creation of shaders by showing the result in real-time. Instead of writing code, every basic shader operation exists as a node which you create and connect in a graph framework. Almost every node also has a built-in preview that enables you to see step-by-step output. How the interface for this looks can be seen in figure 2.4. The graph itself has a main preview, so you can see the end results of your shader at all times, as well as direct application to the materials using the shader, applying the changes to the scene each time the asset is saved.

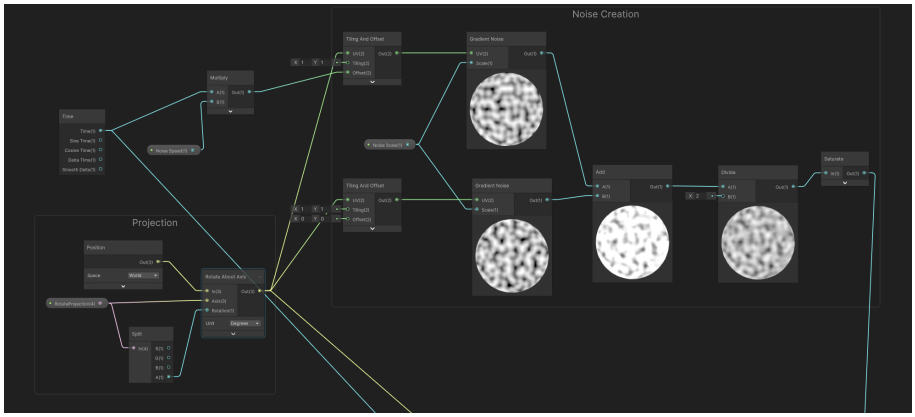

Figure 2.4: Section of a shader created by Unity's shader graph tool. The full shader will be presented more thoroughly further in the report.

2.3.2 Blender

Blender [35] is a free, cross-platform, open-source 3D creation program licensed under the GNU General Public License (GPL)[36]. It contains everything one needs to work on the 3D pipeline. It supports modelling, rigging, animation, simulation, rendering, compositing, and motion tracking. Because of GPL, Blender can keep being a community-driven project in which the public is empowered to make small and large changes to the code base, which in turn leads to new features, responsive bug fixes, and better usability.

2.3.3 Open Broadcast Software

Open Broadcast Software (OBS) or more specifically OBS Studio [37] is a free, cross-platform, and open source application for screencasting and streaming licensed under the GNU General Public License (GPL)[36]. It can capture both sound and video and mix them in real-time. Furthermore, OBS can capture any screen or part of a screen, and handle multiple sources at the same time. Additionally, the captured source can be made to appear as a webcam for the rest of the computer system.

2.3.4 scrcpy

Scrcpy (**Scr**een **C**o**py**) is a free open-source product available under the Apache License, Version 2.0 [38]. Scrcpy allows screen sharing between an Android device and a Windows, Linux, or macOS computer [39]. Additionally, scrcpy does not need to be installed on the Android device, only on the computer. It does work through a wireless connection though it is recommended to use a wired connection.

Chapter 3
Iteration 1

The project was split into two iterations. The first iteration started with defining a concept, followed by a development phase, and ended with the first set of usability tests. In this chapter the process during the first iteration will be presented and motivated chronologically.

3.1 Concept

The initial concept of our setup was to use three devices.

1. An android device running the IFE map-application Arc.
2. A computer running the VR program and handling the communication between all devices.
3. An HMD worn by the user.

The screen in VR would be perfectly aligned with the Android device that runs Arc in reality, showing a screen sharing of the application virtually. That way the user could touch the actual screen and see the effect of their actions in VR. Furthermore, with this solution, there is no need to implement each new iteration of Arc or any other application in VR since you simply use the real system. A drawing of this concept can be seen in figure 3.1.

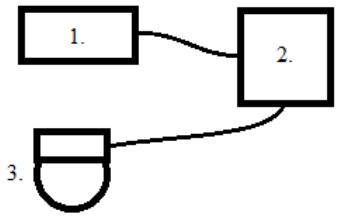

Figure 3.1: A top down view of the different parts of the concept. 1. An Android touchscreen device running the application to be tested. 2. A computer running the VR program and displaying the Android device 1 to 1 in VR. 3. The user wearing an HMD interacting with the screen in front of them and seeing it in the virtual simulation.

3.2 Development

The application was decided to be split into two parts. A start/setup scene where the researchers wear the HMD and can control settings and check that everything is set up for the second part. Which would be the simulation, where the HMD would be worn by the participant and the actual usability testing would be conducted.

3.2.1 The setup scene

Screen Sharing

The first step of performing usability testing in VR is to get the product that should be tested into VR. In this case there were two different options: Either fully reconstruct the IFE application in VR or make the real version visible in VR. As stated in the concept, we would use the second option.

The map application Arc runs on Android devices. Therefore, it would be enough to simply screen share an Android device running the application. Unfortunately, Unity lacks a convenient way to show a connected Android device's screen in VR. Therefore the initial solutions opted to split the problem into two sub-parts. First, screen share the connected Android device to the computer running Unity. Thereafter, show the shared screen in the Unity application. The first step was completed with the help of scrcpy (see section 2.3.4). The second step was solved with the help of Unity's WebCamTexture which can stream the feed from a webcam to a texture in game. This was used in combination with OBS, which, amongst other things, can create a virtual camera that records a specific window on the computer, in this case, the shared screen of the Android device.

The benefits of this solution is the relatively easy and quick development. Unfortunately, it requires the tester to download and run two additional applications which also could induce latency for the user when interacting with the IFE map application.

Aligning screens

As previously mentioned the user would see a screen in VR aligned to a screen in reality running Arc. The process of aligning was done during the setup of a usability test by having the tester define three corners of the real screen and then aligning the virtual world to these three points.

The original plan was to use the HMD's passthrough functionality when defining the three corners. Meta HMDs use cameras to track how the user moves and turns. When passthrough is enabled the feeds from those cameras are passed to the user allowing them to see their real surroundings. This would be helpful when placing the corners as the user could see the screen and where their hands were. Unfortunately, in order to activate passthrough Meta requires the application to run on the HMD, i.e. built to Android. While the WebCamTexture used for screen sharing needed that the application was run on the computer, i.e. built to PC, in order to find and show the stream from the virtual camera controlled by OBS. Which meant that the application had to be built to PC since we did not have a way to stream the screen to Android, which in turn that the passthrough functionality could not be used. Instead, we had to utilise the fact that Meta Quest Pro does not fully envelop the eyes of the user and leaves a gap at the bottom of the HMD. This made it possible to see the real world if you tilted your head slightly backward and looked through the gap, letting us peek at the screen to define the corners. An unconventional and not very sophisticated solution but it worked.

The process of defining a corner was split into two parts. First, the user would press a button in VR which generated a small sphere at the tip of the index finger of the opposite hand, the one not used to press the button. The user would then place the sphere in a corner, beginning with the top left of the screen, by positioning the tip of their index finger there and pinching with their other hand in order to finalise the process, see figure 3.2. This process was then done twice more in order to define the top right and bottom right corners respectively. The final fourth corner was calculated by reflecting the top right corner over the diagonal made up of the other already-defined corners. Once all corners had been defined the user could switch scene to the simulation scene where the usability test would take place. In the simulation, all other objects were then aligned in relation to the screen so that it was placed on the seatback in front of the user.

Figure 3.2: The final step of defining a corner. Positioning the tip of one index finger (left index in image) in the corner of the screen and pinching with the other hand (right hand in image) in order to finalise the process.

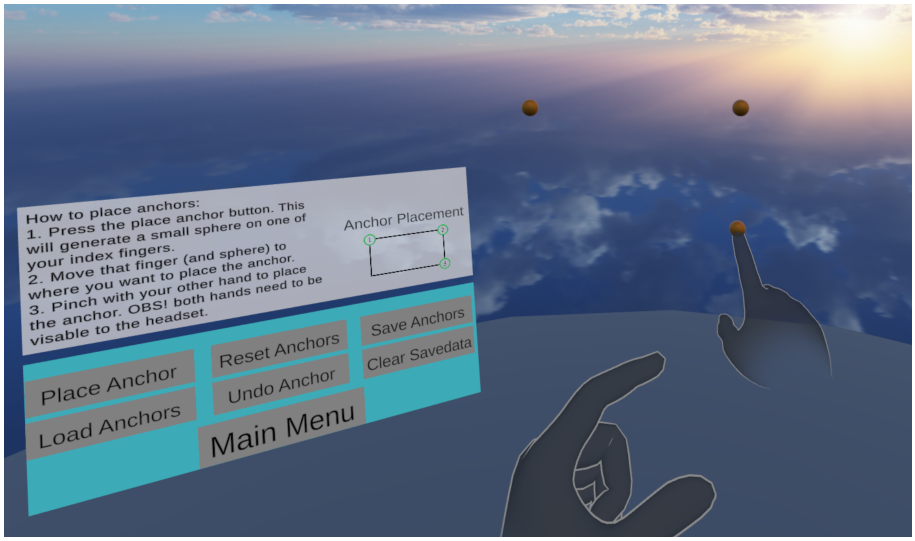

Figure 3.3: How it looks when you are placing anchors in the VR environment. Here the user is placing the last anchor needed, aiming with their right hand and about to pinch with the left to place.

Unfortunately the defined method ran into one critical problem. If the HMD was turned off, or put into sleep mode (automatically done when taken off) the VR world would not necessarily re-align when the HMD is turned back on. And since the testers needed to take the HMD off in order to give it to the users then this solution fell short. However, Meta provides a simple solution, spatial anchors [40]. Spatial anchors are real-world locked points that can persist between sessions. Thus, instead of defining three points in VR space during each setup, the testers would place and save three spatial anchors once. That way when the HMD is handed over to the user or turned off between test sessions the VR world could simply be realigned according to the anchors, which would stay at the corners of the real-world screen. The process of placing anchors seen from VR can be seen in figure 3.3. One

must, however, note that the anchors are not perfect so if the surroundings changed or if the anchors were not in view during the first frame of the application then they could drift, making it necessary to redo the aligning [40]. But as long as the surrounding stays the same and the application always is started while the HMD is facing the real-world screen, then the anchors should be reliable.

Creating the menu system

Even though there is no actual usability testing done in the setup part of the program, how easy it is to set up a test greatly affects the viability of VR as a medium for usability testing. If our VR application is complicated and difficult to navigate during setup then any benefits found during the actual testing might be outweighed by frustration and time costs in the eyes of the researchers. Therefore, the process needed to be streamlined as much as possible, beginning with the menu system. We decided to use panel menus as Monteiro et al. [41] found them to be the most preferred VR menu type. They also stated that the best placement of the menu is on a wall. However, that conclusion was made when using controllers, which can make it easy and intuitive to point and click. With hand tracking however we believed it to be more intuitive to instead press the panel like you would any other button in the real world, and the menus were instead placed floating within the user's reach.

The start menu was simplified by having two layers. The first layer was the *Main Menu* (displayed in figure 3.4), which had ways to directly start the aeroplane environment, quit the application, and move to any of the second layer menus. These second-layer menus were settings, credits, and manage anchors. The main menu also displayed the stream from the currently active camera (see section 3.2.1). When enough anchors had been placed the start button would unlock and the virtual screen placement would be shown. If both the placement and stream looked correct then one could start the testing environment.

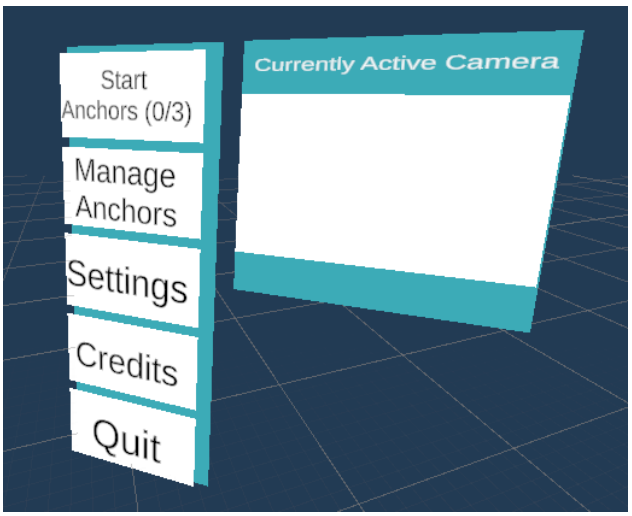

Figure 3.4: The Main Menu of the application. The white empty square displayed the video stream from the currently active camera when the application ran.

The *Manage Anchors* menu, which can be seen in figure 3.5, handled functionality related to placing anchors. In total it contained seven buttons:

1. starts the placement process as described in section 3.2.1 under Aligning Screens.
2. loads anchors from the persistent storage.
3. removes all placed anchors in the current instance (does not affect save data).
4. removes the last placed anchor in the current instance (does not affect save data).
5. saves the currently placed anchors to persistent storage. This would only work if all three necessary anchors had been placed.
6. erases all saved anchors from persistent storage (does not remove anchors from the current instance).
7. returns to the main menu.

The menu also displayed instructions regarding how the anchors should be placed: the first anchor in the top left corner of the screen and then two more in the two following corners moving in clockwise order.

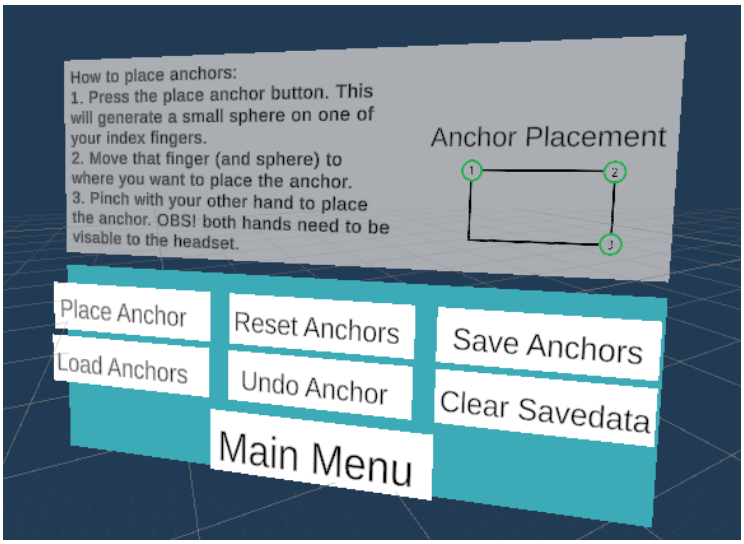

Figure 3.5: The Manage Anchors menu in the application from which one can handle anchors.

The *Settings* menu (see figure 3.6) contained buttons for three different settings and one for returning to the main menu. In addition, the currently active camera was displayed just as in the main menu. The first setting was to choose if passengers should be present in the simulation or not. This was done with two buttons where the button of the selected option would stay pressed until a new option was selected. The next setting controlled what sound should be played in the simulation. These buttons worked the same way as the passenger settings but had three options (off, engines, and engines and passengers) instead of just on or off. The

third setting controlled the active camera, which only had one button. The computer finds all active cameras and then orders them in some order. Pressing the button simply switches to the next camera in that list circling back to the first after the last. Thus, if the computer only detects one connected camera then this button does nothing.

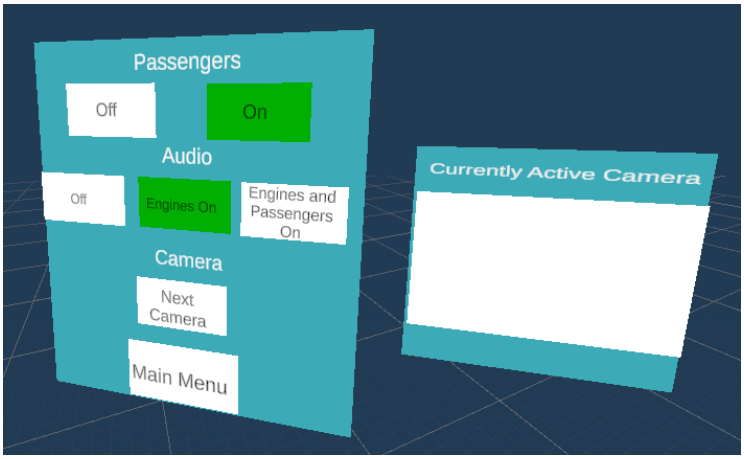

Figure 3.6: The Settings menu in iteration 1 of the application. Here the passengers option *On* and the audio option *Engines On* is selected.

The final menu was the *Credits* menu (see figure 3.7). It only contained one button which would return to the main menu but also displayed credits to the creators of the imported assets used.

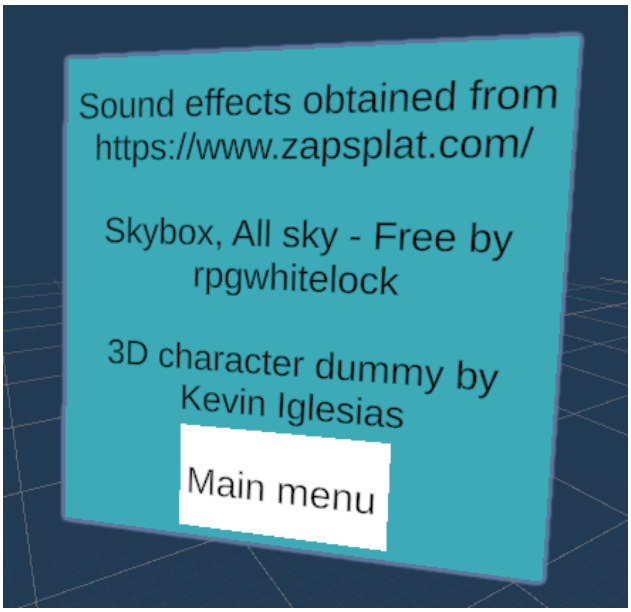

Figure 3.7: The Credits menu of the application.

3.2.2 The virtual environment

Audio

To immerse the user the environment needed to sound like it would from inside of an aeroplane. The challenge was to find a sound that was realistic enough without becoming overwhelming for the participant. The main thing was still to perform a usability test which meant the participant should not be distracted by too much stimulant. We found multiple sound files taken from commercial flights which included the dampened rumble of the engines, dings from the announcement system as well as ambient sound from passengers on board. There were enough shorter audio clips that were sufficiently similar in beginning and end for us to be able to merge them and create a longer audio loop with enough variety where the looping would not be noticed. After deliberation it was decided to create two different sound files, one with passengers and one without. The split was due to our uncertainty over the immersive gain and if it would weigh enough to counterbalance how distracting the sound of talking passengers and screaming children could be. With two sound files we could try different setups and see which fit our test the best. The option with only engines was three-dimensional, with one source placed in each engine. The option with other passengers was on the other two dimensional due to the original files containing both ambient cabin sound and engine sound making us unable to place them in any specific location.

Visuals

To induce a feeling of moving through the sky we needed to create a world outside of the window that moved instead of having the aeroplane "move". When not used in an actual flight the Arc maps application can simulate a random flight between two cities. That meant that if the simulation should to show the world below the plane then it would need to show an image of Earth at the position of the map application to keep the illusion that the plane they are in is the same as the one they see on screen. This task could very well be feasible to do but was deemed more time-consuming than we could afford in relation to how much it would improve the immersion. So we had to create a solution that did not use real-time information but looked realistic.

Our first attempt involved creating a bright clouded skybox by editing images of clouds, merging them seamlessly, and laying some over others at different opacities. Turning this image into a mirror ball and export it into a 3D graphics software tool like Blender [35] meant it could be reconstruct into skybox. To induce the sense of moving through the sky an animation in Blender was created based on the mirror ball, where the layers were interpolated at different speeds between two rotational values around the y-axis. Using different speeds would minimise the repetitiveness of it and make it feel more natural. When finished it looked decently good in Blender's preview but when put into Unity it looked too fake which broke the illusion quickly. Since the skybox was created from only sky images it made it look like Earth did not exist. It did therefore not solve the problem with what the user saw if they looked out of the window.

Instead we tried and settled on creating a clouded landscape below the plane together with a clear blue skybox. Insinuating that the plane is just flying over a heavily clouded area during

the time the participants took their tests. The first step was to create a disc by returning to our 3D tool Blender. The disc needed two things, a large radius to cover the sky without needing to scale it in Unity later, as well as a high polygon count to ensure that the clouds formed from it were smooth. In our case, we created a disc with a radius of a kilometre and subdivided it to a polygon count of 100 000.

Our next step was to create the material that would transform our disc into a flowing cloudscape. To do this we used Unity's shader graph tool [34]. To create clouds of different sizes moving randomly we used noise. Two identical noise maps were created, one static and one offset by a speed parameter. Adding them together, normalising, and saturating them (see figure 3.8) creates a noise with the illusion of having two layers of depth.

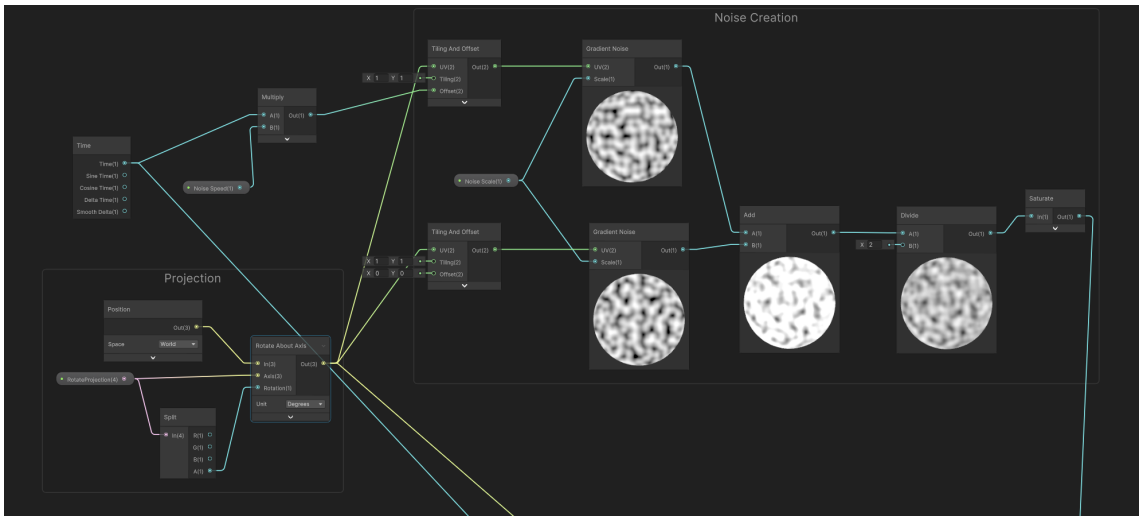

Figure 3.8: Section of the cloud shader graph creating our base noise by mixing static and moving noise

But it still looked too uniform and repetitive to pass as natural. The noise is given as a sine wave between 0 to 1. By remapping the noise to -1 to 1 we say that the difference between a top and bottom is doubled, taking the absolute value of the remapped value will then result in cutting off the "valleys" of the sine wave while still having the output be between 0 to 1. In total this means that we have doubled the number of tops and transformed our valleys into sharper holes as seen in figure 3.9 and 3.10. This enhances the randomness and gives smoother transitions in the cloudscape as we have longer additive parts and short sharp subtractive ones.

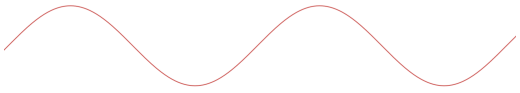

Figure 3.9: A normal sine wave.

Figure 3.10: An absolute valued sine wave. Showing how one can get sharper valleys and doubled the number of smooth peaks.

To make the aeroplane seem to move in a constant direction we could create another noise with simple modifications like strength and offset speed similar to the first and combine it with our heavily modified noise map (see figure 3.11. This base noise then controls in what direction and speed the whole sky disc seems to move in. Our combined noise then acted as a basis for our colour, emission, and geometry calculations (see figure 3.12. For colour and emission we simply add some parameters to the shader, multiply them with the noise and add them to our colour and emission output.

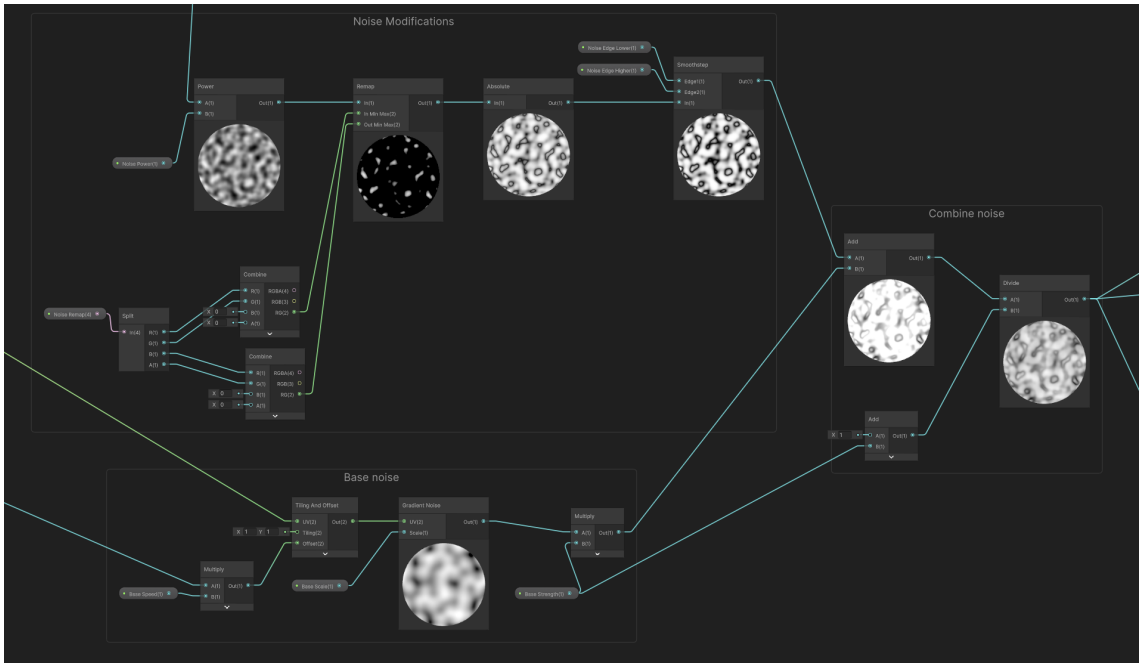

Figure 3.11: Section of the cloud shader graph modifying the two layers of noise into one melded form of peaks and valleys as well as the section creating the base noise controlling overall movement of the clouds.

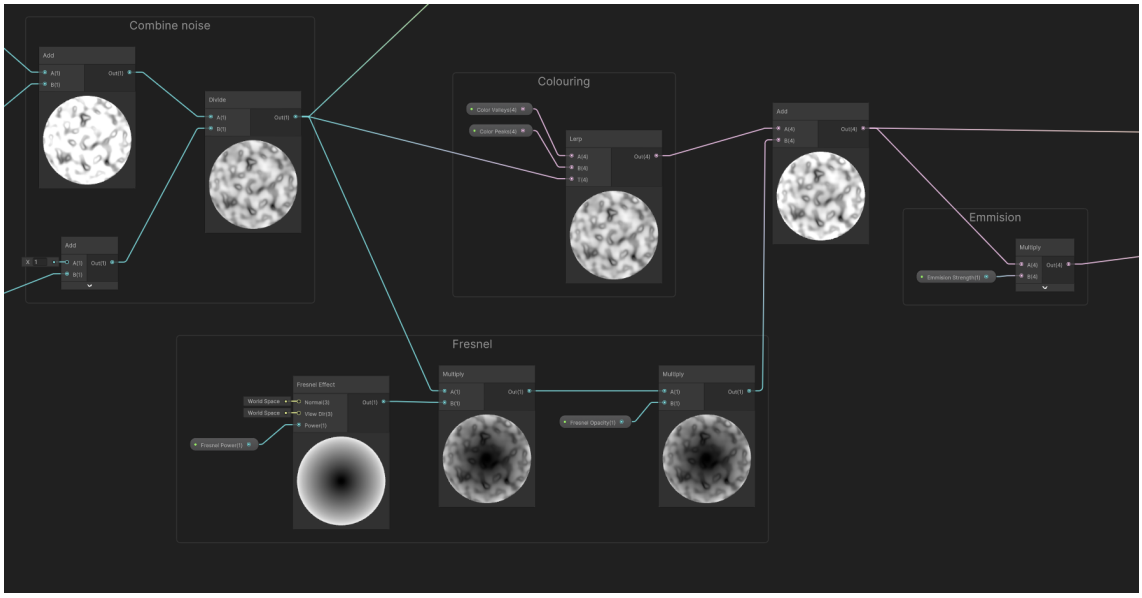

Figure 3.12: Section of the cloud shader graph which uses the combined noise to modify colour and emission of the material.

To actually create the peaks and valleys talked about we used the position and normal of each vertex to get where on the noise map that vertex is, what value it has, and what direction is up for that vertex. Left to do was then only to multiply the vertex up-value with the strength of the noise at that position.

To give the material an even more realistic look three more modifications were added. By adding a Fresnel effect [42] to the colour and emission, the brightness of the material at the edges could be increased, which made the clouds look like they were farther away and went beyond the horizon. In combination with this, we modified the geometry of the disc and created a bowl by rising the edges of it (see *Bowl curvature* in figure 3.13. Making it easier to match where the horizon of the skybox was and let the clouds look more like they stretched toward it.

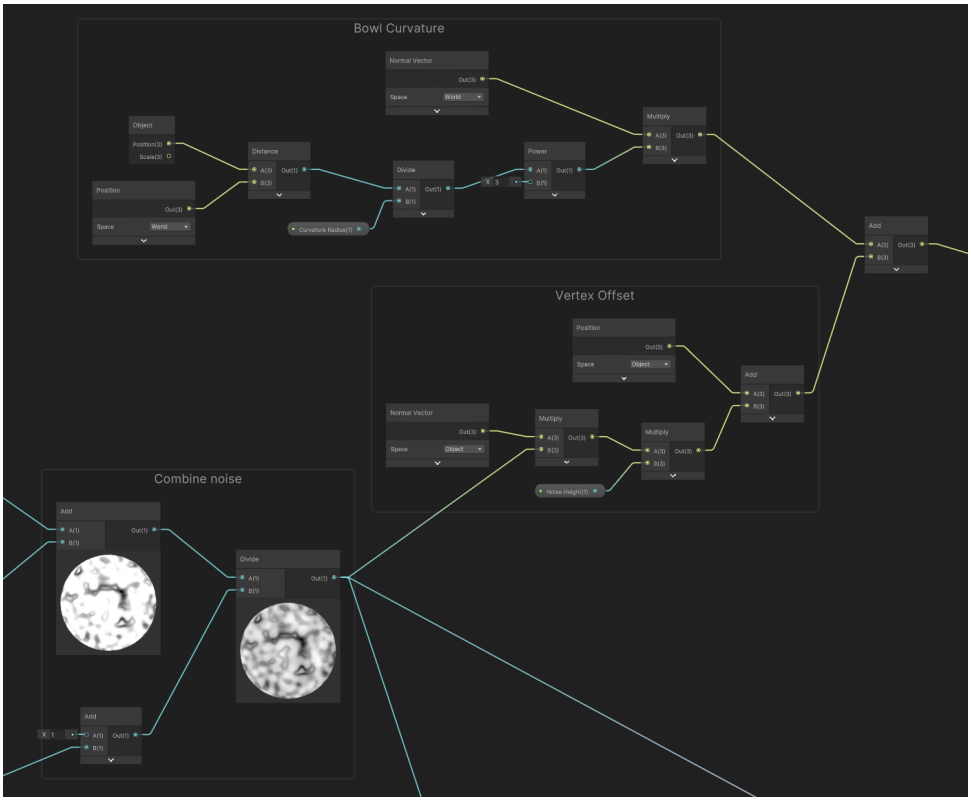

Figure 3.13: Section of the cloud shader graph handling the geometry of the vertices and curvature of the object the material is applied to.

For our last modification we used the vertices screen position together with the camera's depth buffer to determine if a vertex is close to other visible geometry and decreased its alpha value accordingly (see figure 3.13). Creating the visual that the clouds dissipate close to other objects as seen at the closest corner of the cube in figure 3.14.

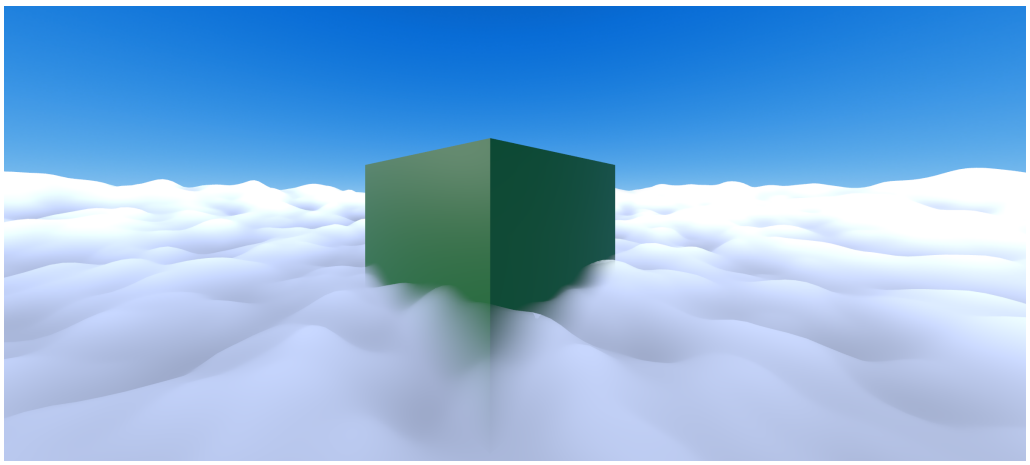

Figure 3.14: The end result of the first iteration of our disk applied with the material using the cloud shader.

3.2.3 Meeting with the design team

In software development, it is always important to keep in mind that, what you as a developer think people want and what the customers actually want, are very seldom the same. Which is why you always should either perform studies or usability tests on your users and keep constant conversations with your customer. In our case, our customer was the design team at Tactel, who were the ones who were going to use our product for their usability tests. A meeting was therefore set up and two representatives of the team were invited to discuss what we had planned and what functionality they needed and wanted.

Device changes

First on the agenda was to discuss how Tactel did usability tests at the time, explain what our initial concept design looked like, and what changes to it were needed to meet their needs. At the beginning of us working with Tactel we had the understanding that most tests were done on tablets running the applications developed by Tactel's different teams. Naturally, we formed our mental concepts around this and started designing a system that used a tablet close in front of the user. However, during the discussion with the design team, we learned that they rarely actually used the tablets themselves when conducting tests. They instead mostly tested either low fidelity (Lo-fi) [1] pen and paper prototypes. Or high fidelity (Hi-fi) [1] prototypes created using design prototyping tools like Figma [43] which ran on a standalone computer instead of the tablets. In the case where they actually used the screens in tests, it was usually for testing the design of the hand controls for the business class setup, which usually meant using an app on a mobile phone as a mockup. There are many differences between business and economy class on a flight, but the main thing that impacts us is how the IFE is set up. In economy, the IFE is a tablet on the back of the chair in front of the passengers. In business on the other hand the screen is usually bigger and further away from the user, more like a tv-screen, which means the user needs a controller to use it.

So the discussion moved to how VR could be applied to any of their existing methods. The design team believed it would not be too much of a change to use the Hi-fi prototypes with touch screens instead, which would work with our already existing design of a medium-sized touch screen close to the user. Additionally, it would eliminate the need for scrcpy in this scenario since the prototypes already ran on the computer. It was thus decided that we continue without scrcpy and an Android device and instead used an external touch screen connected to the computer.

The phone for testing business class controls could work if both sides did some small adjustments. Our problem was that we anticipated it to be difficult to keep the screen sharing in VR stable if a user moved the screen-captured device around constantly. But if the tested screen was modified such that it would not need to be set straight on a wall in front of the user and instead could be tilted, then the design team could create a compromise where instead of the user holding the device freely it would be set fixed in front of them. So it was added to a list of implementations but with lower priority, as the most important part was to make sure normal IFE screen testing worked well.

Creating a more crowded environment

One major thing the designers remarked upon was that they felt we did not make use of the strengths of simulating the environment enough. Foremost it was the lack of making the environment feel cramped. Sitting in economic class on aeroplanes is usually very cramped and annoying with many people sitting close to you. That feeling of annoyance and stress is something they could not recreate in a normal usability test. This led to us exploring adding artificial passengers to our simulation. The discussion with the designers touched on different aspects of these dummy passengers and which were most important. After some discussion, it was clear that the following three questions were the most important to answer.

How realistic should they be?
Regarding making the passengers look realistic it was decided to on purpose keep them more fake due to the uncanny valley effect [44]. If we tried to make them realistic but fell short we instead ran the risk of creating an uncanny valley scenario where the dummies looked close to human but not enough and creep the users out, breaking the illusion. Because of the limited team size of two, it was decided to test going simple in iteration 1 and let the passengers look like crash dummies, see figure 3.15.

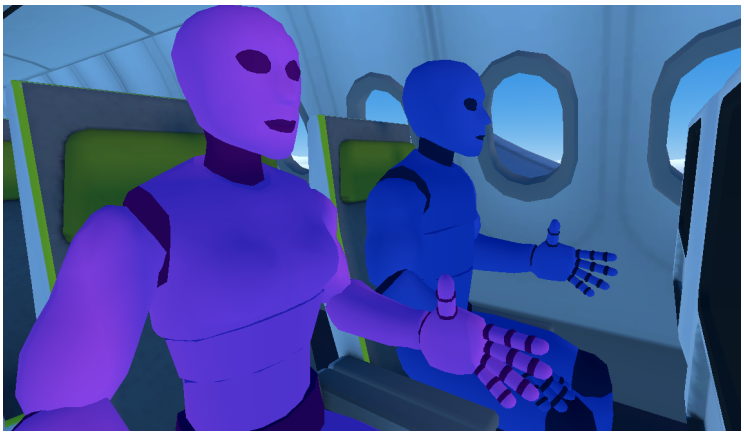

Figure 3.15: Two passenger dummies in their natural habitat.

Should they be animated?
Regarding animation there were multiple ideas on how extensive it should be. When the topic was first brought up a design employee asked about having people moving close to the user, trying to go past to go to the bathroom, and stretching into one's space at times. We were not sure about the idea of something completely virtual trying to squeeze past as it would just clip through the user, potentially worsening presence, it was to the project list of potential features to keep it in mind for later in the process but with very low priority. The discussion turned then towards just having other passengers further away from the user, people sitting restlessly or walking in the aisle. This felt more plausible in terms of believability and could improve the experience of being there with people. The risk however was that it was potentially too big of a task for our team of two. Neither had worked with animations before and were not sure of our capabilities and how long something like it would take.

Figure 3.16: Inside view of the cabin of the Airbus A330-900 Neo model.

In other words, we were not sure that we could create animations in the time we had that looked good enough to not induce that sense of uncanny valley as mentioned before. They would look close to correct but not enough to not make it feel unnatural. Therefore all ideas of animation were put on the list of potential features and saved for later in case there was enough time but also with lower priority.

How crowded should the cabin be?
As mentioned the team felt the the plane environment needed to fully induce the cramped feeling that is so common in especially business class. Therefore it was decided to make sure the cabin was filled to the brim with passenger models. In the potential scenario that the design team needed an empty cabin for some tests, it was kept in mind to group all passengers created so the could easily be hidden later.

Support from the team

The team did also provide huge support to the project by sharing with us a 3D model of the Airbus A330-900 Neo complete with interior design, see figure 3.16. The model had been created for another project at an earlier date by an employee in the design team and needed only some slight modifications to scale to fit the VR environment.

3.2.4 Merging and debugging

At this point in the development's life cycle, it was time to merge the setup and the simulation parts of the program that thus far had been worked on separately. As well as test for and fix any bugs and unwanted behaviour that still existed. While working on the two parts we had always made sure to keep in mind the merge and developed for it, minimising the risk of conflicts. Which worked just as planned, the two parts were merged seamlessly with only needing some references pointed to new assets. All that was left before testing was then to test for unwanted behaviour. As with most programs, this was not one without fault either.

Below is a quick explanation of the biggest errors encountered given, together with a brief explanation of how they were fixed.

After having pressed the button for placing anchors the user was still able to interact with other buttons during the placing phase. This of course, broke a lot of functionality. To be certain that the user did the correct thing of first placing an anchor before beginning another interaction we applied Norman's [13] constraint principle. By temporarily locking all interactable interfaces during the placement phase it was ensured that no unwanted behaviour was possible. We later went on to use this methodology in other areas of the application and temporarily locked buttons while their invoked functions could and should not be called. For example, the save anchor-position button would be locked if there were not three anchors placed in the scene.

Additionally some problems emerged related to handling persistent spatial anchors, i.e. saving, loading, and erasing their saved data. The problems mainly came from these actions needing to be asynchronous [40] and the fact that there is no possibility to simply erase all saved data related to spatial anchors. Instead, one can only erase the data of those anchors that are already loaded and in the scene. The solutions for these problems were relatively straightforward but required an almost complete rewrite of the class that handled spatial anchors. First, a nested class describing a task was introduced alongside a queue containing tasks. Whenever the user pressed a button related to any of the asynchronous actions then a task containing all necessary information and data would be added to the queue of tasks. Then, during each frame, the program would check if it was currently executing a task, with the help of a simple global boolean acting as a lock. If it was not executing and if there was at least one task in the queue then the first task would be popped and performed. If, on the other hand, the application already was executing one of these asynchronous tasks, or if the queue was empty, then the program would simply continue without starting a new one.

Next were the problems with erasing save anchor data. These were fixed by first making sure that all previously saved data was loaded before the user had a chance to interact with anything. Then a dictionary of the currently saved anchors was introduced, with their respective ids as keys. If the user tried to remove one or all anchors (not erase their saved data) then the program would check each anchor that was to be destroyed to see if it was in the dictionary. If that was the case, it destroyed only the visual aspects of the anchors. That way the saved anchors could always be accessed by the erase function when needed.

After merging and debugging a fully functional application was produced. The user's view in the simulation scene can be seen in figure 3.17. Not seen in the picture is how the simulation sounds, which was decided to be the 2D sound file with both engines and ambient passenger sounds. Additionally, in figure 3.18 an overview of the application's scenes, alongside each of their main areas of responsibility, can be seen.

Figure 3.17: User's point of view from their seat in the final product from this iteration.

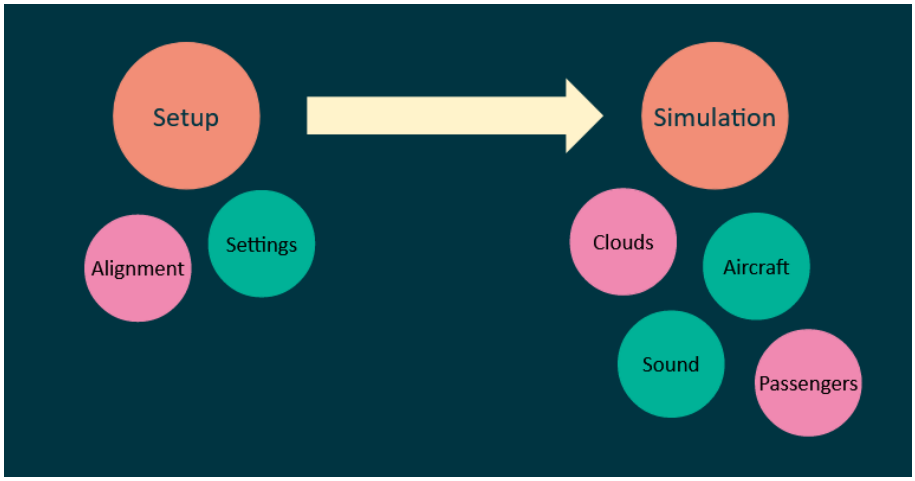

Figure 3.18: Overview of the application's scenes and their respective areas of responsibility.

3.3 Testing

3.3.1 Test plan

As described in section 2.1.1 the test plan is a core part of usability testing as it acts as the blueprint of the whole process, ensuring consistency between tests. With only a team of two, there was less emphasis on using the test plan to explain to other departments how the test was to be conducted. Instead, the main function of the test plan was for us to go back and double-check and remind ourselves in case we forgot any parts of the test.

Purpose

The purpose of this test is to determine how doing a usability test in virtual reality (VR) impacts the results both in terms of objective efficiency, as well as the subjective attitude of the participants towards the test. To compare the participants we will use counterbalancing and split them into two groups. One doing their tasks first in VR and then redoing them in real life and the other group doing it in the reverse order. The test flow that this creates can be seen in figure 3.19. There one can also see the three tasks in each test and their main areas of interest. The tasks will be further defined later.

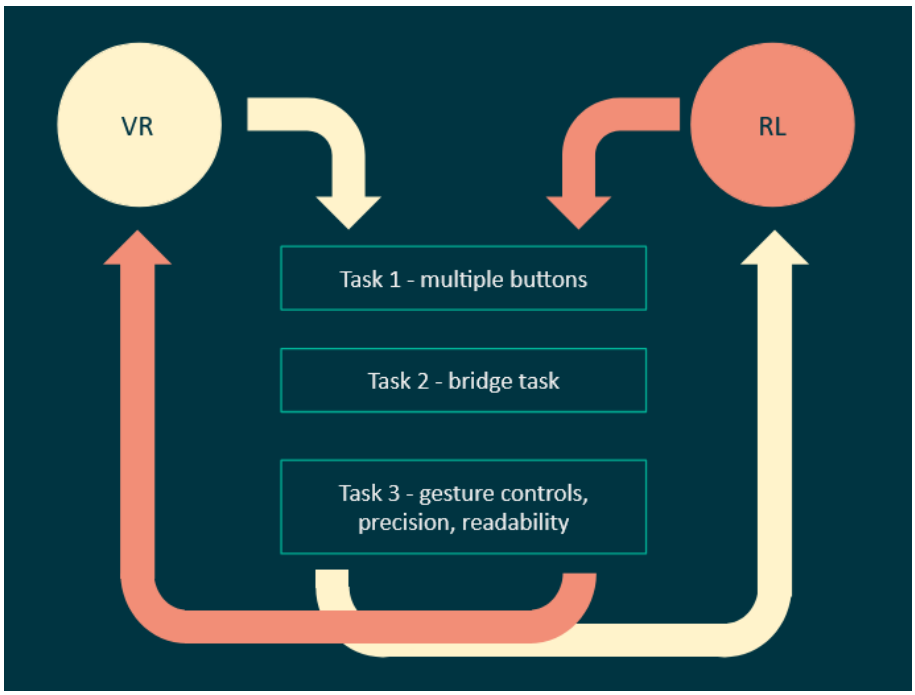

Figure 3.19: The test flow during a test. Half the participants started in VR while the other half started in RL.

Research Questions

- How does the result of the usability test get impacted by being in a virtual environment?
- How does the experience of doing the test get affected by being in a virtual environment?
- Is the VR controlling scheme intuitive and easy enough to use so as to not be a hindrance for testing, and to what extent?

Selection of participants

To ensure privacy and full anonymity participants will never be referred to by name in any official presentation but instead be called participant #X. Due to administrative limitations, the tests will only be conducted with participants from the office at Tactel and close family and friends of people working for the company. This will limit how broad of a user group we can find in terms of age and previous experience with both VR and inflight entertainment systems. Our participants were between the ages of 20-65 and ranged from medium to high technical knowledge. In terms of VR experience we know less beforehand but expect a spread between "never used" and "At least once a week".

The test would as mentioned use counterbalancing where half of the participants did the first part in VR and the second part in real life and vice versa. This meant it needed enough participants to get satisfactory results from each usability test and double it for counterbalancing. Due to time restrictions, we still needed to keep the participant count as low as possible and looked at getting 5-7 participants for each group meaning a total of 10-14 participants. Without administrative limitations however the test could be carried out with any willing participant regardless of flight, IFE or VR experience.

Data to be collected

The different types of data are grouped into four categories by combining Objective and Subjective with Quantitative and Qualitative data.

Table 3.1: Data collection table

RQ	Objective Quantitative	Objective Qualitative	Subjective Quantitative	Subjective Qualitative
#1	Task success, Time taken, Number of errors	Analyse performance		Debriefing interview
#2			Presence questionnaire	Debriefing interview
#3	Time taken, Number of errors	Analyse performance	Presence questionnaire	Debriefing interview

Task list

Table 3.2: List of tasks in the order in which the participants perform them during the test. Each task contains certain subtasks that describe how the task should be done, a success criteria for the moderator to determine when they are done with the task and a maximum time each task is allowed before the moderator should move on with the test

Task	Subtasks	Success criteria	Max time
Check distance left to destination	Change from imperial units to metric in the settings menu. Go back to flight information and check the distance left	The participant can tell the moderator how far it is left to travel measured in metric units	2 min
Check out the "aircraft view" tab	Open the main menu and change to correct tab	The application is in the tab "aircraft view"	1 min
Get more information about any city	Search the map using gesture controls until a city with an info i is found. Press the info button by the city on the map and press read more on the info bar that comes up	The application is on a page where the user can read more information about a city and the user can read a brief paragraph	2 min

Procedure

Table 3.3: Phases of the test procedure in order of operations. Includes what action is done in each phase, material needed to perform the actions as well as the maximum time each phase should take

Phase	Activities	Materials needed	Time
Briefing	Orient participant Sign consent	Orientation script Informed consent Pre-test questions	4 min
Test 1	Perform task in task list	Scenario/task descriptions Observation protocol	5 min
Switch	Reset program Add/remove HMD	Meta quest pro	1 min
Test 2	Perform task in task list	Scenario/task descriptions Observation protocol	5 min
Debriefing	User fills in questionnaires Post test interview	Presence ques- tionnaire Post test questions	15 min

The total time needed for the whole test procedure was estimated to be maximum 30 minutes. To be ready for potential technical issues or other delays, 45-minute time slots were booked with each participant. With a 15-minute break between participants for correcting notes and resetting the program, participants were estimated to take an hour each.

Test environment and equipment

All tests were carried out in the experience-room at Tactel Malmö office. This location gave the test the most privacy and non-distracting environment mimicking that of a real lab room the most. By being made up of multiple smaller nodes of tech, the setup was constructed simply and modular enough so that if the need arose the whole test environment could be moved to a different location.

Equipment

- A Meta Quest Pro head-mounted display
- A Quest link cable
- A computer meeting the requirements for running Meta
- A device for the participant to answer questionnaires on
- A device for each observer for note taking
- A device for measuring the time taken
- A tablet or touch screen, preferably wall mounted
- Cables needed to connect tablet/screen to the computer
- A chair for the participant, preferably high backed as to match the simulation

Test environment
The test environment is set up in a similar fashion as that of the "simple lab setup" described in Handbook of usability testing [12]. The participant is placed in a high-backed chair in front of the touch device with the HMD. The moderator is placed to the left slightly behind the participant so as to be just in view and contactable when the participant is not using VR but not so close that they interfere or stress them. The observers and tech-responsible testers are placed as far back as possible to minimise their disturbance while still being able to see what the participant is doing during the test. The computer running the simulation is placed with them and it is therefore helpful to get a longer set of cables so they can be drawn around the testers and participant and be out of the way. An overview of the setup used in this study can be seen in figure 3.20.

Depending on if the screen used to simulate the chair-back IFE is a tablet or a touch screen, there are two different ways to set up the test environment. If the team uses a touch screen then that screen is simply connected to the computer which in turn extends its desktop to the screen. The computer then runs both the VR simulation as well as the application to be tested and has the application on the now second monitor, the touch screen. OBS is then set to record the whole second screen and stream it to the simulation directly. If on the other hand, the team uses a tablet then the tablet itself runs the application to be tested. The device is still connected in the same way to the computer but instead of just capturing the device as a second monitor you instead run scrcpy together with OBS as described in **Screen sharing** under section 3.2.1.

Figure 3.20: Overview of the test setup with moderator and participant to the right and observer/tech to the left. Disclaimer, in this image the screen used in iteration 2 can be seen in the image, the only reason for that is that no picture of the setup was taken when conducting our first tests, in reality, it was not there at this point in the study.

Division of work

Due to the small team size, there were only two roles:

Moderator: Handle the participant during all phases of the procedure. Read task scenarios to participants, observe, and keep track of time.

Observer/Tech: Monitor the application and fix any potential technical issues, observe and take thorough notes during testing. Important to note that the person with this role is also the only one seeing the application and therefore what the participants see in VR. This means that it is important that they are keen on observing their performance.

If the test is run with a bigger team then it is a good idea to break up these two roles into smaller ones that only take care of one or two things each, letting the tester have a better focus on their task, increasing efficiency.

Observing and note taking can be difficult to juggle. During tests, there is often a lot of information coming at you all at once. What the participant says, what they do, how the program reacts, and so on. How you as an observer behave in the test room can also impact the participant and by proxy the result of the test. To minimise that risk a set of guidelines published by the Nielsen Norman group were followed where they were applicable to this project. [45]. To decrease the observers workload one could instead record the test, putting less pressure on them seeing everything. However, due to the nature of our tested program which needed two points of views, one external that sees the participant and one internal that sees the VR environment we saw problems with recording everything. Without

a proper laboratory the recording would also have needed to be done on the same computer running everything else already. Which we were afraid would be too much for it and impact the performance of the simulation and map application. Therefore we decided to try without recording in this iteration and determine our need for it afterwards.

3.3.2 Pilot Test

Before actually running the test we needed to make sure that there were not any mistakes made in the planning that would force us to fix things while wasting the participants' time. A pilot test was therefore conducted with our office supervisor acting as a participant. Their participation was deemed as solid evidence of a real test as they had not seen how the simulation was at that point in time nor seen anything of the test material beforehand.

From the pilot test we found that our simulation, test plan, and questionnaires all worked as planned. This of course did not mean that there would not be any misunderstandings or questions from real participants later, but we knew at least that there would not be any test-breaking disturbances. Only two things in the test plan were changed after testing. First off it was noted that it was easier for the moderator if the orientation script, scenarios, and task list were printed out on paper and that they had a separate timekeeping device instead of everything being on the same laptop. The scenario description was also shortened the second time it was told as we felt it dragged on to hear the whole thing again and got the same reaction from the participant.

3.3.3 Test

During the test, the participants were encouraged to use the Think aloud method which is a great way for us to better understand how they think and why they perform their tasks in certain ways. The method, unfortunately, runs the risk of skewing the results as the scenario becomes even more unnatural than it maybe would be in a real-life setting [46]. People are not used to voicing their own thoughts and some could struggle with talking and thinking about what they are doing. Especially when they get stuck, which is when we need to know what they think the most. This problem could potentially also be enhanced by being in a more stressful environment which a test certainly can be. Even more so if something goes wrong. It often helps to let the participant envision having a friend or colleague beside them that they explain the tasks to [12]. Therefore, before beginning the test we let the participants know that we wanted them to think out loud and said that they could pretend the moderator sitting next to them was a colleague.

Participants

A total of ten employees from the office participated in the study. When asked what they identify themselves as three answered men and seven women. As seen in figure 3.21 their age varied mostly between 26-35 with some from other age groups.

Every participant answered that they had "Tried or used VR a select few times (not regularly)", nine people said that it was from testing playing VR games and one participant chose

to not say what they had used it for. These numbers were expected to be heavily influenced by the fact that the office had a VR headset that people could borrow over the weekends and try. In other words, many of them had tested the very headset we were using for our study in their free time.

7/10 participants answered that they worked with inflight entertainment systems. Out of those seven, it is unclear if they had worked with the exact application tested. But as we will see later in the results their knowledge about the application did not impact our study's result in any major way, i.e the group of participants tested were deemed correct such that the test would not need to be redone with a new set of participants.

We also asked the participants how often they fly (see figure 3.22) and when they do how often they use an IFE system (see figure 3.23). This was in order to see how accustomed they were to flying and how much experience the participants had with IFE applications in the product's natural environment.

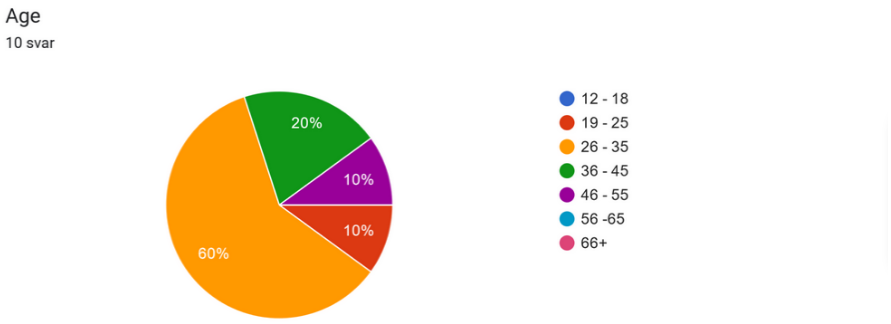

Figure 3.21: Age distributions of participants.

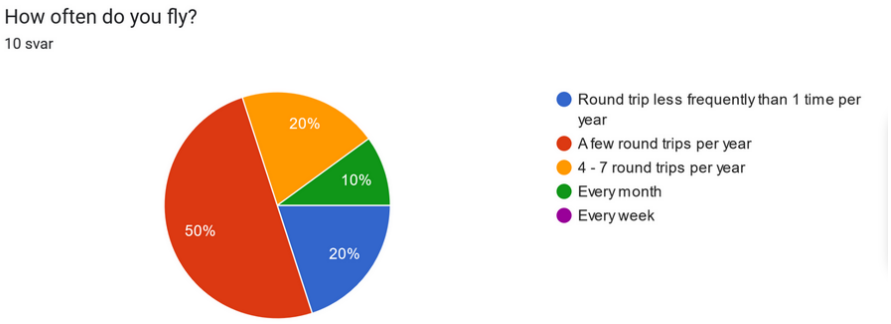

Figure 3.22: How often participants travel by plane.

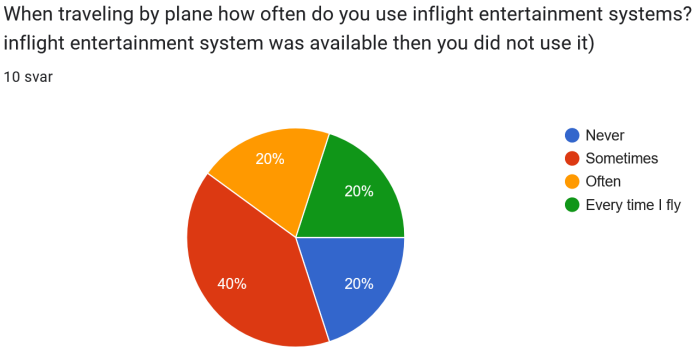

Figure 3.23: How often participants use IFE systems while flying.

Tasks

As mentioned the test was broken up into two parts, one with VR and one without. All participants did the exact same tasks described to them in the exact same way by the moderator in both parts. Only the scenario description was told differently in the second half as described earlier. The participants were asked to perform three tasks of varying difficulty in precision. The first task involved going through multiple menus to determine how well tapping multiple buttons of varying sizes worked. The second task was a shorter bridge task setting up for task three as well as testing how quick and easy actions were affected. Which leads to task three, which handled taking in a lot of information, trying different gesture controls, such as rotating and zooming a map as well as testing the smallest buttons to press, the information i:s over cities on the map as seen in figure 3.24. This task also involved the most differently sized texts for us to determine how the resolution would impact the readability of applications. The exact tasks read to the participants can be seen in appendix B.3.

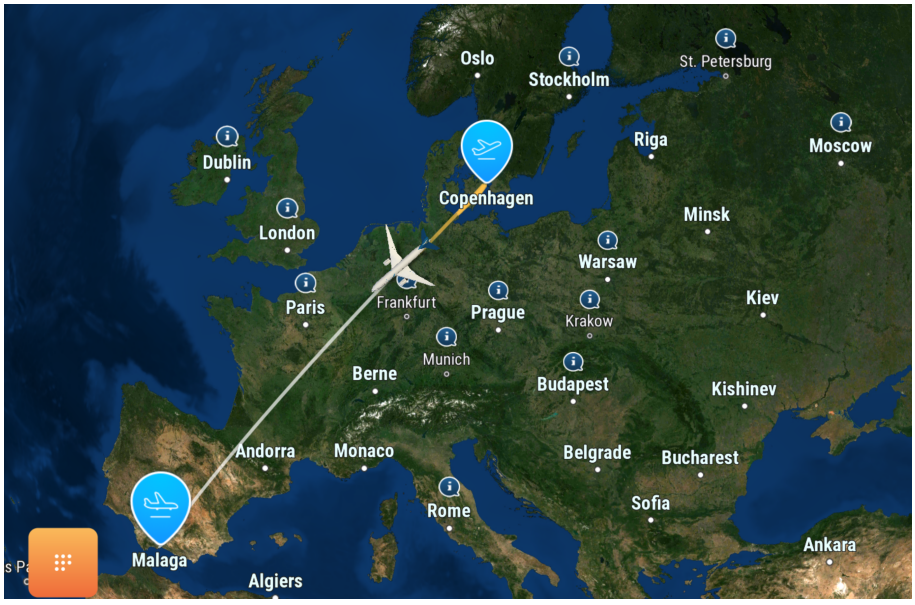

Figure 3.24: The map application to IFE systems that was used in the usability tests.

Debriefing

Wrapping up testing we followed up with a series of more in-depth questions which focused heavily on their subjective feelings about the experience. Beginning with a predetermined base set of questions. Instead of trying to predict everything in beforehand we decided on a more conversational approach and let the discussion form the questions asked instead of vice versa. Because of this technique the debriefing naturally evolved during the test procedure as the topics were explored with different participants. As testing went on new important questions were gradually discovered that we had not thought of beforehand which were added to the set of base questions asked to other participants moving forward. The base questions started with can be read in appendix B.3. By adding questions later in the process we of course missed the earlier participants' answers to these questions which could impact the conclusions drawn from their answers later. We therefore continuously kept this in mind while analysing the results and always double-checked how they had answered and compared to which questions they were asked as everything was recorded by the observer.

After the interview all participants filled in our presence questionnaire (PQ) which was used to determine how engaging the simulation was and how high each participant's sense of presence was. Our presence questionnaire was built on the questionnaire first put forth by Witmer and Singer [20] but with a few questions removed due to them not working in our case and some changed to better fit our scenario. The resulting PQ used can be found in appendix B.4. As mentioned earlier in section 2.2.1 under *Measuring Presence* the PQ on its own is not sufficient to determine a person's presence. However, combining it with our observations during testing and the participants' answers during the post-test interview should allow us to more objectively determine each participant's presence. These scores will be discussed in the results analysis section below.

3.3.4 Results

Below are the results of the first usability tests presented. The section is categorised by tying the results found with their corresponding research question. If some data is relevant for multiple research questions then they are presented in the section of the first applicable and later just referenced.

How does the result of the usability test get impacted by being in a virtual environment?

Task completion time varied greatly between participants and is presented in table 3.4 and figure 3.25. Participants with an odd number started in VR while those with an even number started in RL. As one can see, in general, the tasks took longer to complete in VR.

Table 3.4: Time taken by each participant on each task (T 1-3) both in Virtual Reality and Real Life as well as the total time taken for each participant and the average time for each task.

Participant	T1 VR (s)	T2 VR (s)	T3 VR (s)	T1 RL (s)	T2 RL (s)	T3 RL (s)	Total time (s)
1	56	7	19	24	7	36	149
2	68	17	62	28	6	60	241
3	77	14	26	26	4	27	174
4	63	9	19	76	11	22	200
5	46	8	23	13	3	10	103
6	74	10	18	56	3	89	250
7	63	5	132	21	2	14	237
8	174	12	36	63	5	62	352
9	130	13	100	44	5	27	319
10	29	3	52	97	4	46	231
Average (s):	78	9.8	48.7	44.8	5	39.3	225.6

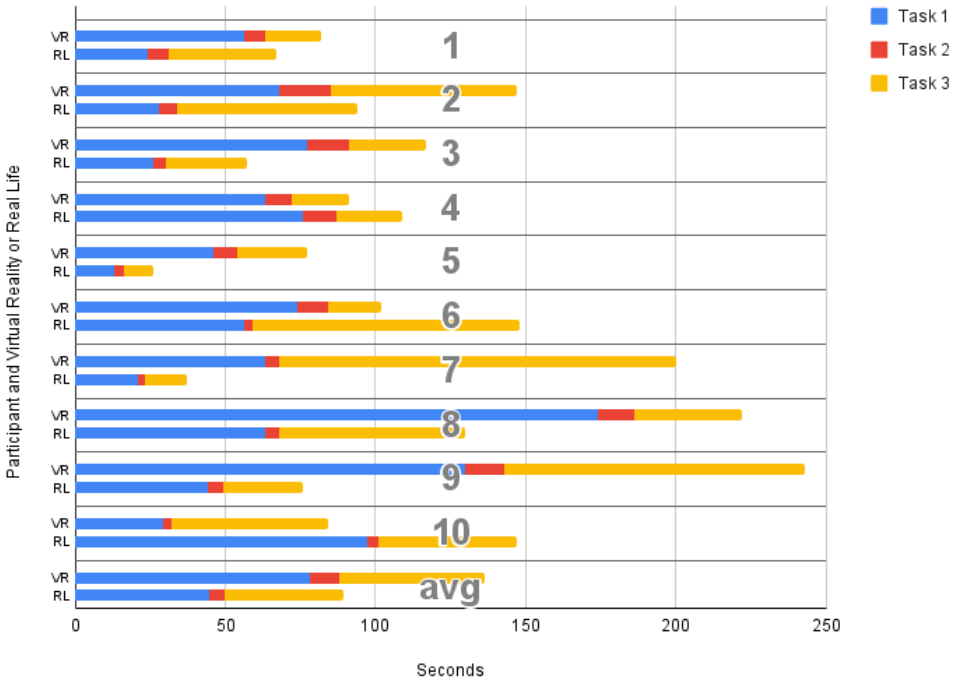

Figure 3.25: Task duration for each participant, both in virtual reality (VR) and in real life (RL).

Eight out of ten participants expressed that there was some type of problem related to hand tracking or the accuracy of their fingers. Seven of those said it was when interacting with the screen whilst the eighth just felt it was generally off. The remaining two commented that the hand tracking felt good and accurate. However, each participant had at least one instance during the test where the hand tracking caused them to miss a button, regardless of how they said it was. Additionally, all participants encountered at least one instance where

they struggled to read some text. How difficult it was varied, most participants found the smaller text slightly more difficult than normal whilst others found it cumbersome or even impossible to read. These difficulties, misses, and other moments when the participant was performing the correct action but VR caused them to fail will from here on be referred to as *VR-related drawbacks* or simply *drawbacks*. In iteration 1 we only saw the two types of drawbacks previously described. Instances of these VR-related drawbacks can be seen in figure 3.26. Important to note is that we counted failing instances and not every time something occurred. I.e if a participant missed the same button multiple times in a row or complained about the same text multiple times then that would only count as one instance. However, if a user gave up due to not succeeding to press a button and got prompted by the moderator to try again, then that would count as one additional instance.

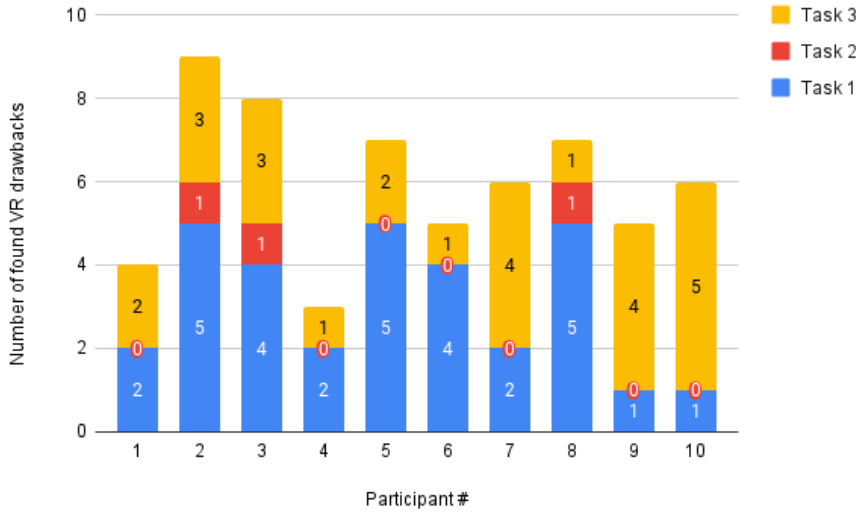

Figure 3.26: Instances when a VR related drawback was found. In iteration 1 a drawback instance was counted either when the hand tracking caused the user to miss a button or when the resolution of the HMD hindered text from being read.

How does the experience of doing the test get affected by being in a virtual environment?

As mentioned earlier, the hand tracking affected the participants to various degrees. Four out of ten participants expressed or showed clear signs that the inaccuracies of the hand tracking had been frustrating or irritating. Three participants said that the hand tracking was good or even equivalent to their real hands. The remaining three participants could not clearly be placed into either group.
Figure 3.27 shows the average score for each question in the PQ that is in some way related to hand tracking. These questions were questions 2, 3, 5, 15, 17, 18, and 21 and read as follows:

- **PQq2**: How natural did the interaction with the environment seem?
- **PQq3**: How naturally did you actions impact the visual aspects of the environment?
- **PQq5**: To what extent did your hands in the virtual environment feel like your real hands?
- **PQq15**: How proficient in moving and interacting with the virtual environment did you feel at the end of the experience?
- **PQq17**: How much did the hand tracking controlling scheme interfere with the performance of assigned tasks or with other activities?
- **PQq18**: How well could you concentrate on the assigned tasks or required activities rather than on the mechanisms used to perform those tasks or activities?
- **PQq21**: How easily did you adjust to the hand tracking controls used to interact with the virtual environment?

The average is displayed separately for the participants who started in VR to those who started in RL. The different questions were scored on a scale of one to seven where, usually, seven means high realism or presence and one relates to no realism or presence. The exception amongst these questions is PQq17 which is a negative question and thus the scale is reversed, i.e. a lower score is considered higher presence.

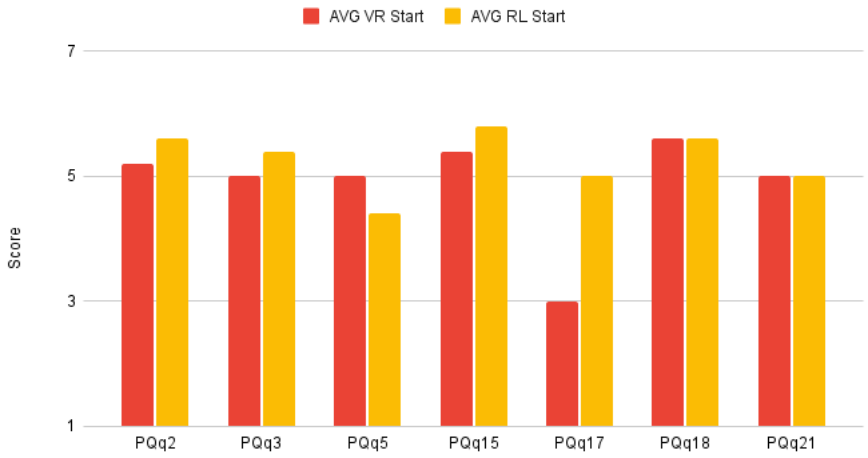

Figure 3.27: The results of the questions in the Presence Questionnaire that were in some way related to hand tracking. The questions were scored from 1 to 7 where 7 usually relates to higher presence and 1 no presence. PQq17 however is a negative question and thus that question's scale is reversed.

There were parts of the simulation that the participants did not feel were up to the same standard as the rest. Seven out of ten participants commented on the passenger dummies in a negative way. Them being blue and sitting completely frozen made most participants find them unrealistic or distracting.

The results from our presence evaluation of each participant can be seen in figure 3.28. There it can be concluded that eight out of ten participants had a medium (4) or higher presence. Note that our presence evaluation is scaled from low (1) to high (7) presence and not from no to full presence. In the same figure, each participant's presence questionnaire score is also presented. The PQ score was calculated as

$$\frac{1}{24}\sum_{i=1}^{24}\begin{cases} x_i, & positive\ \ question \\ 8 - x_i, & negative\ \ question \end{cases}$$

where x_i is the participant's answer to question i. A positive question is a question where a higher answer corresponds to higher presence. For example *PQq12* seen in figure 3.29, was a positive question. Negative questions on the other hand are questions where a lower answer corresponds to higher presence. For example *PQq13*, seen in figure 3.30, was a negative question.

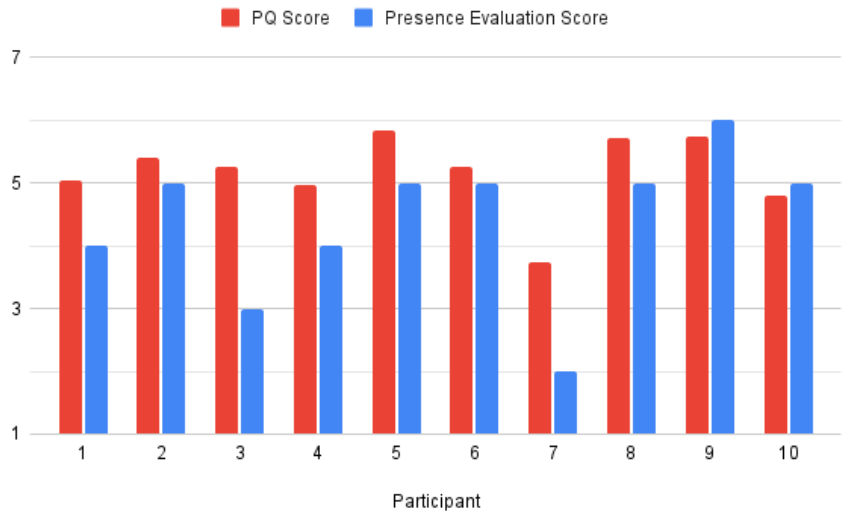

Figure 3.28: Each participant's PQ score alongside our presence evaluation of the same participant. Note that a discrete scale was used for the presence evaluation.

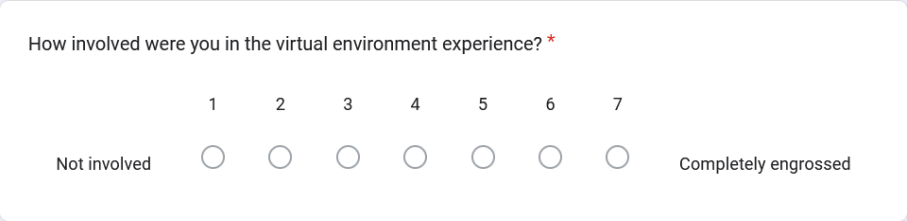

Figure 3.29: Presence Questionnaire question 12. An example of a positive question.

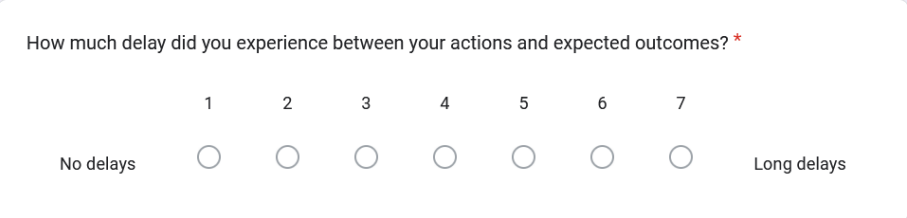

Figure 3.30: Presence Questionnaire question 13. An example of a negative question.

As mentioned earlier, all participants at some point struggled with reading text due to blurriness. This does not only affect the result of the usability test but could also affect the overall experience for the participants. As reading blurry text could not only be tiresome and annoying but also decrease presence as the user knows that the text should not be that bad.

Overall, doing the test in VR seems to have been appreciated by the participants. Nine out of ten participants commented that doing the test in VR was either more fun, more interesting, or both. Four participants mentioned that the simulation put them into more of the correct mindset. They felt that they were actually out travelling and got the excitement and positivity that came with that mindset. A summary regarding the participants' general opinions regarding the test as a whole (attitude), regarding hand tracking, and their presence, can be seen in figure 3.31.

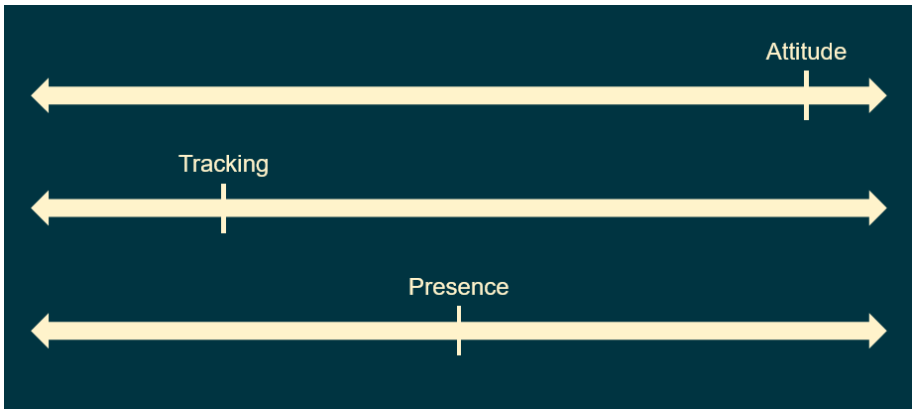

Figure 3.31: A summary of the participants' opinions regarding a few key aspects in regards to the tests done in VR in iteration 1. Right is more positive and left is more negative.

Is the VR controlling scheme intuitive and easy enough to use so as to not be a hindrance for testing, and to what extent?

All participants tried to touch the screen with their fingers without us having to explicitly explain that they had to, regardless of whether they started in VR or RL. However, it should be stated that due to a misunderstanding with participant four which caused them to be surprised that the screen in VR was the real screen, it was decided to explain that the screen in VR was the real screen to later participants. Even though the explanation given did not mention how to use the VR hands, it could have still made the participants realise how to interact with the screen in VR. Unfortunately, as stated earlier, the inaccuracies of the hand tracking still caused multiple instances where the participants missed a button. Meaning VR did become a hindrance in those cases.

3.3.5 Result Analysis

In figure 3.26 we can see that the minimum amount of drawbacks found was three and the maximum nine. Task one was the longest (as in, most buttons needed to be pressed in order to complete the task) while task three required the most precision. Therefore we expected more instances of drawbacks in task one, but that most participants would have more difficulties with specifically pressing a city in task three. However, many participants showed to have adjusted to the inaccuracies by the time they reached task three, either consciously or automatically. Participants four, six, and eight experienced no accuracy-related drawbacks in task three in VR and as such, since they all started in RL, have faster completion times in VR for task three. Also participants one and three were faster in VR for task three even though they started in VR. This is however most likely due to other factors, such as exploring the application more the second time around (in RL). On the other hand, participants seven, nine, and ten had many more accuracy-related misses in task three than in any other task, though all for different reasons. Participant nine started by choosing a city without additional information and thus had to switch to a different city. Both selections requiring a few presses but never enough to annoy or clearly affect the participant. Participant ten had close to no problems with task one, only one button requiring a second try, but for some unknown reason seemed to be far off in task three. It was almost as if the HMD had lost part of the hand tracking accuracy between the tasks. This confused the participant as things were fine before. Finally, participant seven used another method of pointing at the screen. Every other participant tapped the screen with their hand down and finger straight up, letting the camera clearly see the finger. Participant seven however tapped the screen with both finger and hand perpendicular to the screen, obscuring the finger and disturbing the tracking. When the tracking felt off the participant tried to adjust their hand to fix it. The problem was that their idea to point better was to have the hand up and point the finger downwards, which did not solve the occlusion problem. This led to further frustration and later a lower opinion of the experience. We also opted to not explain how the hand tracking works to the participant which showed how frustrating things can be when they do not work.

It is clear from figure 3.25 and table 3.4 that the tasks generally take longer time in VR. The primary contributing factor to these time increases were found to be the inaccuracies with hand tracking. In cases where the hand tracking worked and was accurate VR seemed

equivalent to RL from a control scheme viewpoint. On the other hand, when things did go wrong, the time taken to solve that error was closely related to whether or not the participant could identify why things went wrong. If they could not then things took a lot longer, but if they figured out that the errors were caused by the hand tracking, then we saw only a small amount of additional time taken. The participants did still find it generally harder in VR to achieve the desired outcome, mainly due to said tracking inaccuracy.

Looking at figure 3.27 we see that for the most part, there is little to no difference between the opinions related to hand tracking between those who started in VR and those who started in RL. The only exception is PQ question 17, (*How much did the hand tracking controlling scheme interfere with the performance of assigned tasks or with other activities?*) where those who started in VR felt the hand tracking interfered a lot less. This could potentially be due to those starting in RL knowing that the screen worked very well when used in RL and thus knew that the only reason for the screen to not work in VR would be hand tracking inaccuracy. While those who started in VR might have felt that it instead was problems with the screen itself. It is not clear which case is better. On one hand, blaming the screen rather than the hand tracking would indicate a higher sense of presence, but blaming the product for an error that it does not have could be troublesome for usability testing.

In regards to our presence evaluation score, seen in figure 3.28, we noted an overall medium or higher sense of presence across the majority of participants. We began evaluating the results by looking at occurrences where the participants struggled with the application. If they tried to solve their problems using reasoning that would apply in RL i.e. that it was the screen or map application that was wrong, then that was seen as an indication of higher presence. For example, participant four believed the blurriness in VR to be due to them not wearing their glasses or contacts. That is even after they had already seen the screen in RL and could read it fine then. Another example is participant eight who pressed harder repeatedly in the same spot which indicated that they thought the touch was unresponsive, whilst, in fact, they were actually one centimeter to the left of the button.

Also contributing to their presence evaluation score were the participants' post-test interview answers. If a participant expressed feelings of being in an actual aeroplane or that their hands in VR were their real hands then that would be indications of higher presence. With opposite answers indicating lower presence. Additionally, if something distracted them from the experience, such as the dummies being seen as unrealistic then that would also indicate a lower sense of presence. Also worth mentioning is an important point concerning our evaluation, specifically related to participant seven. This participant showed multiple signs that could be interpreted as both high and low presence. After deliberation it was decided to interpret it as low but note for further analysis that the result is ambiguous and could have been misinterpreted. That being said, in general, we saw relatively high senses of presence, both in regards to our evaluation and in regards to the average PQ score. These scores could be influenced by the fact that we had a connection to all participants since they all worked at Tactel. The participants were prompted to answer the PQ and our questions as truthfully as possible and not what they might think we want to hear, though the results might have been different if the participants had not know us in beforehand.

To summarise: the main drawback with VR identified from our tests was inaccuracies with hand tracking. Hand tracking inaccuracies not only affected the participants' hands but also the screen placement. Since placing anchors that defined the screen were done by pointing with ones index finger, it was also dependant on the accuracy of the hand tracking. The other drawback was blurriness though it seemed to have had a lesser effect overall. The participants seemed mostly positive to usability testing in VR even though these drawbacks occurred. There did also not seem to be any difference in what the participants commented on or how many things they commented on regarding ARC when in VR versus in RL. From these results, we cannot say that VR seems particularly useful for usability testing though they do show what parts need to be improved.

Chapter 4
Iteration 2

The second iteration mimicked the parts of the first while skipping the no more needed concept phase. The main difference between iterations was the shorter development cycle in iteration 2, which only went on for two weeks. Creating a project and getting to a point where it felt like reliable tests could be conducted was a much bigger task than improving on what we now had. Important to note however is that the testing and analysis got the same amount of time as in the first iteration

4.1 Second meeting with the design team

Being done with our first iteration cycle it was time to have another meeting with our "customer", the design team. So far we had only discussed features together and described what we were working on orally in short meetings and during breaks in the office. Three from the team joined us for a meeting. One had had a short test of the simulation previously when they discussed preparations for the usability tests with us, but it was still in development at that point and missed a number of features. For the other two, this was the first time they experienced the product altogether. In great news for us, they were very impressed by the program.

To be able to accurately determine if VR is viable to use in testing there was still an important aspect needed to test. Can new testers learn to set up and use the application without needing too much guidance and time? If the tests could not be replicated easily then all data gathered so far only worked because it was us who did it, which decreases the viability of using VR for testing. To test this we prior to the meeting wrote a simple instruction manual on how to set up each part needed for the application to work. This included the external screen, OBS, the Meta HMD, the IFE application used in the tests, and of course the simulation itself.
After some initial discussions about the product and a brief showcase of the simulation itself, we turned everything off, gave the list of instructions to the visiting team, and let two of them cooperate to set everything up. There was no need to do a whole test with them one

by one as they would probably not be doing usability tests alone anyway in the future and instead opted for a more realistic scenario. Also, listening to their discussion would give us more insight into where the problems lay, working like the think-out-loud method but in a more realistic setting. The two participants started with both reading the instructions and doing the initial parts of setting up OBS together and later switched to having one of them use the VR headset and the other read the instructions and used the computer.

Everything during the test went smoothly except for one thing, placing the anchor points in VR. We expected that to be the hardest part as how you position the headset and place the finger greatly impacts how well the camera sees where your fingertip is. So after letting them try and fail by themselves first, we offered some advice on how to place the anchors. Not predicted however was that even with our help and seeing them do what looked like the correct thing, the screen created by the anchors was still wrong. A phenomena noticed before but thought only of as an outlier. Fredrik himself always struggled with placing the anchors as, regardless of how well he did the actions, the screen created would always be slightly wrong. But if Rasmus placed them in the same way then it worked as intended. At the point of writing this we still do not know what caused these issues, it was first believed that the camera just did not like Fredrik's hand for some reason but seeing another person get the same problems weakened that simple theory. Because we did not know why this happened to some, we decided to make it a priority in the next iteration of development to make the program instead fit the best rectangle to represent the screen out of the anchor positions and known screen size. Hopefully limiting the problems with placing the anchors.

Even though there were problems with the anchor placement, the team was overall very happy with how smoothly it went to set up and saw great potential in continuing working on this project. They felt the perks of VR, i.e the whole experience of being in the correct environment and the possibility to gain new currently unavailable metrics like eye-tracking, have the potential to heavily outweigh many of the negative aspects, e.g taking more time to set up, users VR inexperience impacting the tests, additional costs, etc. There were some concerns over the problems with the accuracy of the tracking and the flickering text which they also experienced when testing the simulation. If they persisted then they could negatively impact tests too much. However, they were aware that our study is more of a proof of concept and that the problems found here could be worked on with more time and planning.

As an example, the last thing discussed was how much the size of the screen impacts the resolution of text on the screen. Technically the whole screen is affected but the text is simply the most noticeable. The size of the screen in VR directly affects its own resolution due to how many pixels of the HMD it covers. We could easily verify that the text on the screen became easier to read if we increased the size of the screen in VR. However, since we were reliant on the virtual screen being mapped one to one to the real one the size of the screen in RL then directly affect the resolution of the screen in VR. The company seemed to already have some plans on buying an additional touch screen so one interested designer took on the task of getting a screen some inches bigger to see how much of a difference it would make.

4.2 Development

From the first iteration there were three main things that needed to be improved:

- the alignment and positioning of the simulated screen,
- a touch indication for when one miss-pressed in VR,
- and the other passengers.

Additionally, some small adjustments had to be done to accommodate for the bigger screen that we were able to use for the second iteration.

4.2.1 The simulated screen

As seen in iteration one, inaccuracies from hand tracking were one of the leading factors to drawbacks. We could not improve the accuracy of the hand tracking itself, but we could limit its effect on screen placement. The previous method of defining the screen resulted in a high potential for errors in each corner. These errors would combine as one moved toward the screen's center resulting in different errors across the screen. Our attempt to improve this was to use the screen's real size as a parameter in combination with a least square algorithm in order to align the screen. This method would not eliminate the errors of hand tracking when placing the screen, but it would limit them and spread them equally over the whole screen, making them more predictable.

The actual implementation of this method started by having the user place four anchors, instead of the previous three, one in each corner of the screen. Next, a least square method was used with these four anchors to find the best fitted plane according to:

$$\begin{pmatrix} a \\ b \\ c \end{pmatrix} = (A^T A)^{-1} A^T B$$

where a, b, and c were the parameters of the plane equation $ax + by + c = z$, A was a matrix that for each anchor had a row containing its x and y value as well as a 1, and B was a column vector containing all the anchors' z values.

Next the anchors were projected onto the calculated plane and the center of the screen was placed in the middle of the projected anchors. The screen could then be rotated to align its normal to the plane's normal, with the assumption that the screen lies perfectly horizontal. This change barely affected the set-up of the simulation, only that one more anchor had to be placed. In figure 4.1 the new menu hub after all anchors have been placed is displayed. Note the anchor in the bottom left corner of the right screen and the start button's *4/4* instead of the previous *3/3*.

Figure 4.1: View of the menu hub where testers set up the simulation for the participants. Now with the updated requirement of four anchors placed.

4.2.2 Touch indication

Another way to try to counteract the tracking inaccuracies was to add a way for the user to see where they actually tapped the screen. We saw multiple instances during the tests where the participants missed the button they intended to press and often did not know how they missed it. This led to them needing to guess in which direction the virtual finger was wrong. Sometimes they guessed correctly whilst other times they did not, leading to more aimless tapping on the screen. The more they failed their task the worse it became as their irritation mounted. By not giving correct and instant feedback to the user we violated a core design aspect, one which is part of Norman's list of design principles [13]. The aspect being of course Feedback, that every action needs a reaction and that the user should always be notified if the program is working based on it responding to their actions or if something went wrong.

To solve this we needed to have some indication that showed the user where they actually pressed so that they could correct it correctly. The touch screens used for the tests did have a faded circle that showed each time one tapped the screen. The problem was that the built-in indicator ran on the screen itself which meant that OBS could not capture it, i.e. it did not show in the virtual camera and therefore not on the screen in the simulation. For it to be visible we needed an indicator that ran either in the Unity application or on the computer the screen was attached to. Due to time restrictions, we decided it was not worth trying to implement our own from scratch when there are good tools already developed. For simplicity, we instead chose to use Microsoft PowerToys [47]. PowerToys is a set of freeware system utilities designed for power users on the Windows operating system. It includes multiple different tools but the one thing we were interested in was the *Mouse utilities* tool. The mouse utilities included an option to turn on cursor highlighting, creating a circle around the mouse whenever you pressed or held the button. This worked well for us as tapping a touch screen connected to the computer is the same as moving the cursor there and pressing the mouse

button, i.e. with this tool active a pink circle showed at the spot one tapped. Applying a one-second fade to it so it lingered a bit after you had moved your hand made it more visible. All that was needed now was to test to see if participants used it to correct themselves or if they just misinterpreted what the circle meant and became more confused by it.

4.2.3 The passengers

From the test, it was noted that the participants exhibited a strong dislike for the passenger dummies used. People said that they were a bit weird and many believed it would be better if they were more lifelike. This was most likely due to the uncanny valley effect mentioned before. We thought the crash dummy look would seem non-human-like enough to not be weird, but it did not, they were still too human-like which just made it worse. To improve the experience of the simulation the passengers needed to be changed to get rid of the uncomfortable feeling it gave participants. There were two ways to do this. Either we tried to make them more lifelike, which seemed a very hard task to do with the time remaining if we wanted to ensure that uncanny valley effect was fully removed. Or, we tried the opposite and made the passengers even less lifelike on purpose, by turning them all into a model that fits being fake. The best fit for this description was found to be a robot model. It would look reasonable enough as to why it was in the setting and detached enough from any human resemblance to remove the uncanny valley effect. We found a robot model in a Unity creator starter pack that fit our purpose well. It only needed some slight modifications to scale as well as some posing to fit the scene. Using the new model two containers of passengers were set up in the simulation scene. One for robots and one for dummies. The containers were filled with models sitting in the aeroplane and could easily be toggled on and off to display different sets of passengers. This would later be used during the test to ask participants their opinions on these different sets.

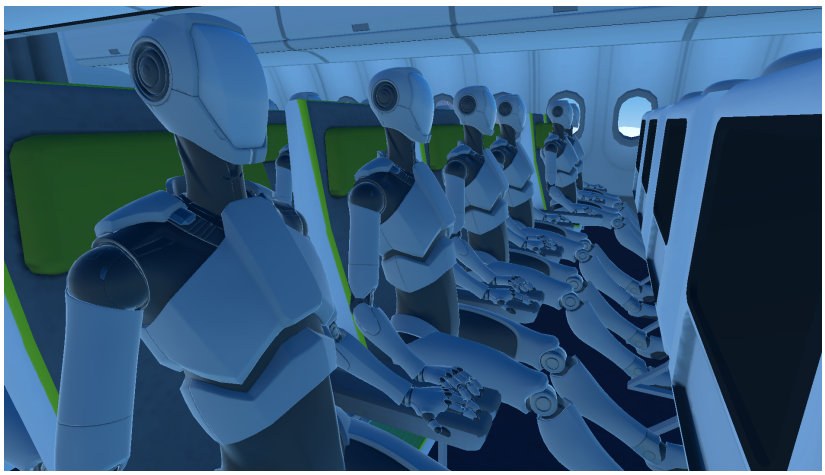

Figure 4.2: The robot passenger model posed in the aeroplane.

4.2.4 A bigger screen

For this iteration, we also got hold of a bigger touch screen to test on. As mentioned we wanted to test if having the streamed texture take more space in VR solved the resolution problem and the flickering text. The new screen measured 17 inches across i.e. five inches bigger than the previous one.

Figure 4.3: Point of view for a participant in the virtual environment with the new screen.

4.3 Testing

4.3.1 Changes in iteration 2

Orientation script and tasks

Even with modifications to the program, the best course of action for testing in iteration 2 was to keep the original test plan and our scripts mostly the same. In part so the results from both tests could be compare to see if our improvements made any difference, but also because it was deemed a strong basis already. There were however some improvements made to the script that we felt only enhanced the test without changing it too much.

First the second shorter scenario was removed and replaced it with the same longer scenario the test began with. We originally created the longer scenario to give the participant a more detailed description first and the shorter one to just use it as a reminder of the scenario the second time, so it would not be taking too much of the test's time. We found however that since we counterbalanced our participants with half beginning in VR and vice versa, having two different scenario descriptions meant that those who started in VR did not get the longer description when they needed it in RL to paint a picture of the scene. We also could not just simply switch so that the longer description always was for the RL part as it was confusing to have a shorter excerpt scenario in the VR part if you began there. The solution found was just to have the longer be read both times.

As mentioned in the development section we added the possibility to switch between passenger setups. This allowed us to add the question of preference where we cycled between the different setups and asked the participants their opinion on them and which they preferred. Another aspect noted during the first iteration was the participants' answers to the PQ questions regarding how well they could identify and localise sounds. Even though the sound used was completely 2D (the same no matter where the participant turned or moved) the answers still indicated that the participants thought they could localise where it came from. Therefore, to enhance the presence questionnaire questions "How well could you identify and localise sounds", the participants were also asked to tell us what sounds they heard and point to where they came from whilst still in VR.

Some of the post-test questions were improved upon and some new ones were added based on the extra questions that often came up in the first iteration. As there was times where we either needed to rephrase a question to explain it better or ask extra ones to delve deeper into a subject. All changes to the orientation and task descriptions can be seen in appendix B.3.1.

When analysing our results in iteration 1 we found that most of the time our notes taken during the tests were thorough enough such that no information was missed or lost. Sometimes however, we felt we were missing the tone and intention of the participants' answers. To fix this a recording device was brought to the post-test interview this time and the participant were asked if they were okay with their answers being recorded. If they did not agree then the post-test interview would simply continue without any recording, like in iteration 1.

Participants

So far, one of the bigger weaknesses of this study was the lack of diversity in its pool of participants. Due to the recruitment restrictions stated the application was mostly tested on employees of the company. Many of them had seen the map application "tested" before and knew the actions needed to complete their tasks. In our case it was not too bad because of the nature of this test where we do not care too much about how well the tests are done but instead just the difference between RL and VR. To diversify the test participants in this iteration we sought to recruit people from outside of the office. To keep the process simple these participants were all known to us in beforehand.

Test setup

The test setup was kept mostly the same between iterations. The main change was the screen used, changed to the new bigger screen. A stool in the corner by the screens was added to pose as a stand for a recording device (laptop) during the post-test interview. We used the same computer for recording as for participants to fill in their screening form and the presence questionnaire, which meant no addition to the list of devices used. The resulting test setup is displayed in figures 4.4, 4.5, and 4.6.

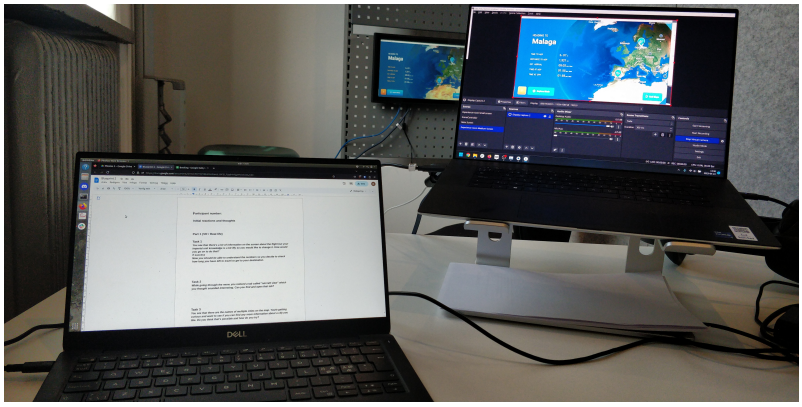

Figure 4.4: Overview of the test setup from the observer's point of view.

Figure 4.5: Overview of the test setup, how the moderator and participants sit during the test.

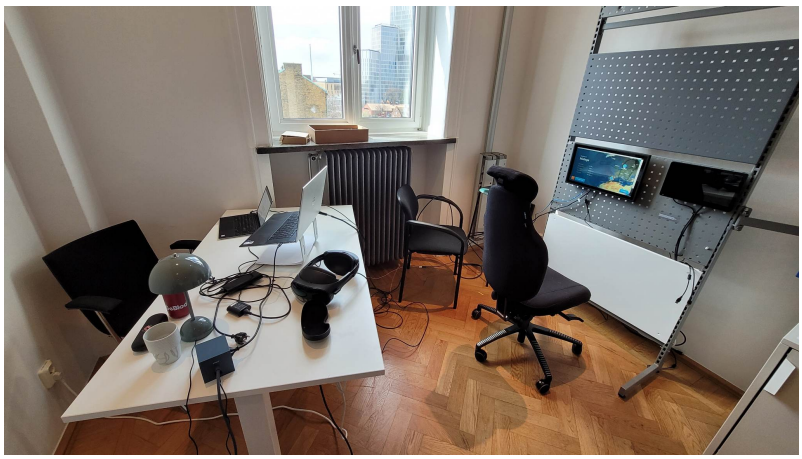

Figure 4.6: Overview of the test setup from the side, showcasing the whole room. During the test a wall was put between the screen panel and the observer table, crowding the participant to improve the aeroplane feeling.

4.3.2 Test

Participants

For this round of testing, we got ten new participants who had not seen the project beforehand. Unfortunately, there were difficulties recruiting people who had time to come to the office in Malmö during work hours. In total we got four participants who were fully external, two who were master thesis workers on the company like us, with some knowledge about the system but not as much as the employees, and lastly four employees. Which is fewer externals than hoped for, but still a better spread than none, as in the first iteration. Every participant was new and had not seen the test before. In terms of gender identification, we got a better spread this time with five men and five women. As seen in figure 4.7 most participants were again aged between 26-35 but with the average age being lower this time.

VR usage was somewhat more diverse but mostly the same as in iteration 1. Two had never used VR before. One answered that they had used to use VR a lot but did not currently at the time of the test. The rest answered as in iteration 1 "Tried/used it a select few times (not regularly)" with each of those having "played VR games".

The participants were once again asked how often then fly (see figure 4.8) and when they do, how often do they use an IFE system (see figure 4.9). Compared to iteration 1, the participants in iteration 2 were a lot less accustomed to flying.

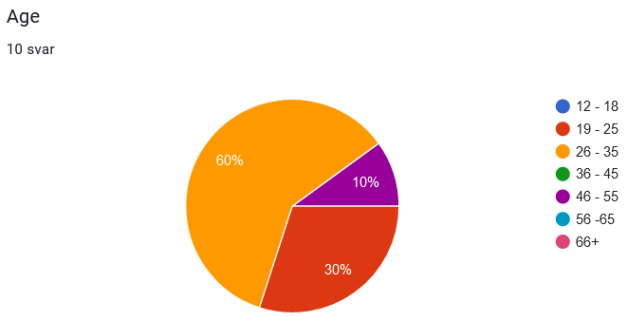

Figure 4.7: Age distribution amongst participants.

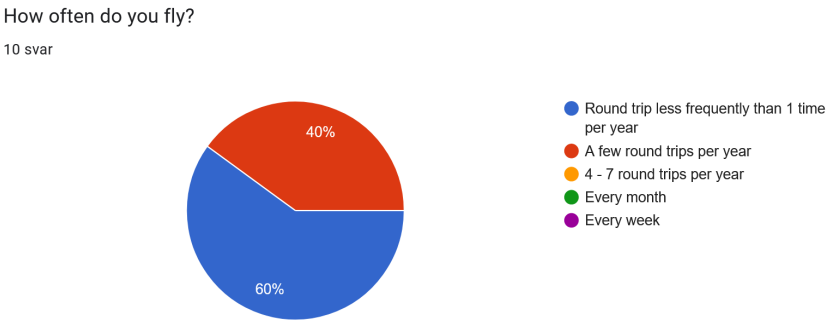

Figure 4.8: How often participants travel by plane.

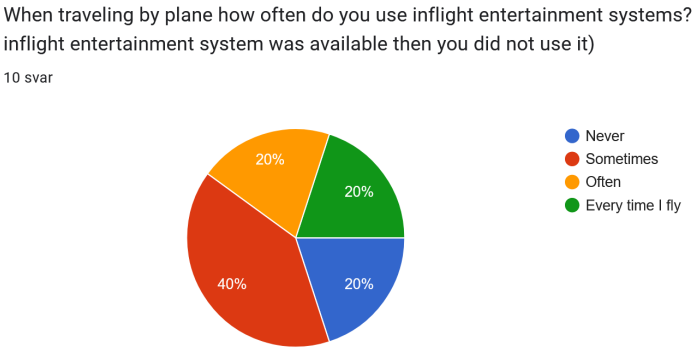

Figure 4.9: How often participants use IFE system when flying.

4.3.3 Results

Below the results of the second usability tests are presented. The results found are categorised by tying them with their corresponding research question. If some data is relevant for multiple research questions then they are presented in the section first applicable and later just referenced.

How does the result of the usability test get impacted by being in a virtual environment?

Test completion time varied mostly between 160 and 200 seconds with three participants being significantly faster and two being slower. These completion times as well as each task's completion time are presented in table 4.1 and figure 4.10. In general, the participants were slightly faster in RL. As in the first iteration, participants with an odd number started in VR while those with an even number started in RL.

Table 4.1: Time taken by each participant on each task (T 1-3) both in virtual reality (VR) and real life (RL) as well as the total time taken for each participant and each task.

Participant	T1 VR (s)	T2 VR (s)	T3 VR (s)	T1 RL (s)	T2 RL (s)	T3 RL (s)	Total time (s)
11	80	13	61	24	5	12	195
12	37	7	32	29	5	23	133
13	43	9	27	38	8	58	183
14	71	6	28	63	5	20	193
15	95	6	19	84	4	34	242
16	57	5	84	60	10	120	336
17	56	5	42	40	2	30	175
18	23	9	15	72	7	35	161
19	37	3	26	22	2	17	107
20	18	3	22	37	2	23	105
Average (s):	51.7	6.6	35.6	46.9	5	37.2	183

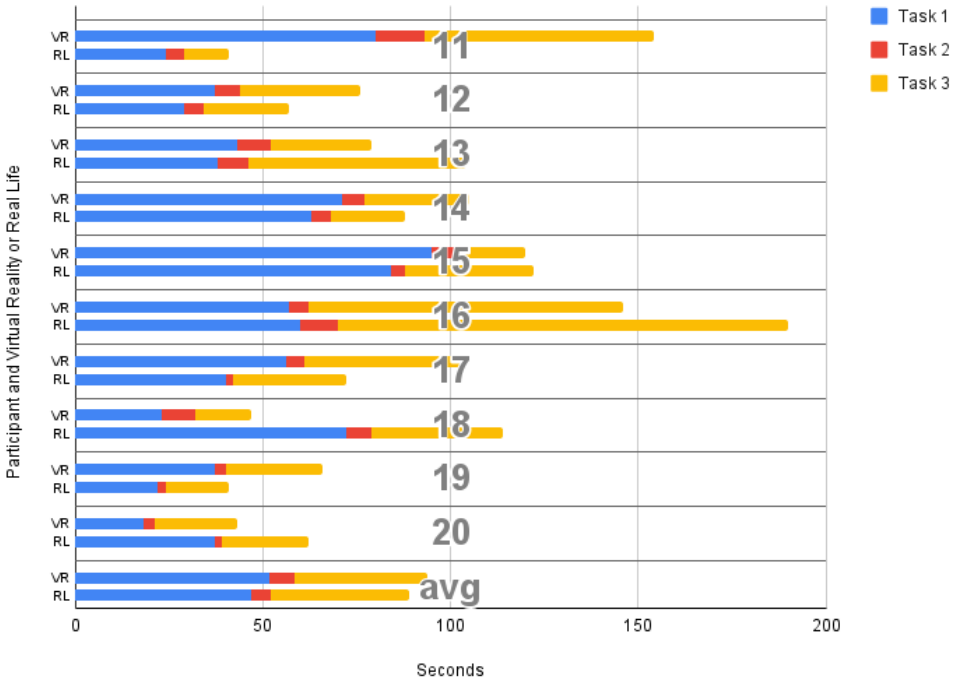

Figure 4.10: Task duration for each participant, both in VR and in RL.

Only two out of ten participants expressed that the accuracy was in one way or another bad, but no participant described it as frustrating or irritating. Four out of ten described it as slightly off but still okay, and the remaining four described it as correct or natural. Most participants still experienced some accuracy-related drawbacks. Participant 15 however had no such drawbacks and participant 16 only had one miss in total which was solved quickly enough for them to not notice the touch indicator implemented to aid in these scenarios. In

general, each instance of accuracy-related drawbacks was solved quicker and had less amount of misses per instance in this iteration.

Accuracy related drawbacks as well as other instances of VR-related drawbacks can be seen in figure 4.11. These drawbacks were measured in the same way as in the last iteration, i.e that a drawback instance was counted whenever a participant made an error that was caused by VR and not themselves and that if a participant missed the same button multiple times in a row or complained about the same text multiple times then that would only count as one instance. Only four out of ten participants expressed that some text was difficult or impossible to read. However, in this iteration we also noticed instances where the screen suddenly blinked, sometimes multiple times in a row. If the participant commented on this blinking then that was also seen as a drawback. This drawback was only commented on once, more specifically for participant 16 in task 3.

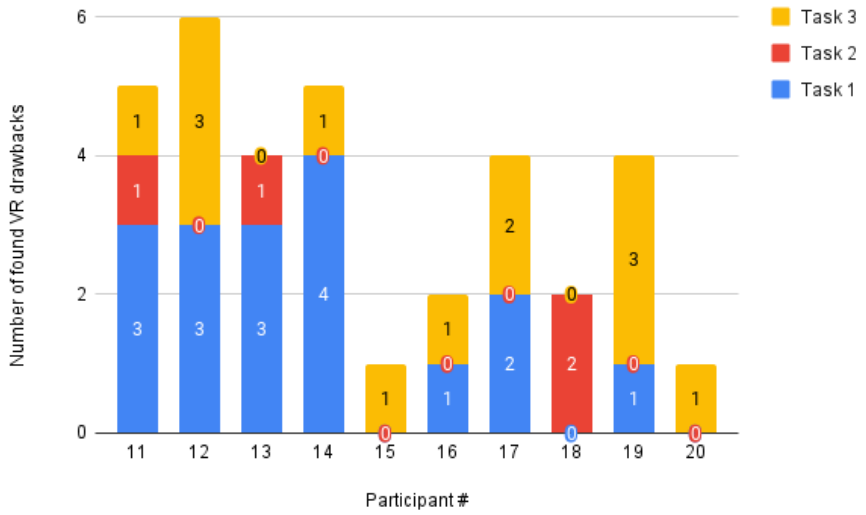

Figure 4.11: Instances when a VR related drawback was found. In iteration 2 a drawback instance was counted when the hand tracking caused the user to miss a button, when the resolution of the HMD hindered text from being read, or when the screen started blinking enough to distract the participant.

How does the experience of doing the test get affected by being in a virtual environment?

Just like in iteration 1 we also looked at the PQ questions related to hand tracking (see section 3.3.4). The average answer to these questions for those who started in VR and those who started in RL are presented in figure 4.12. Nine out of ten expressed the hand tracking in general as good or that they did not think about it (i.e. close enough to real life as to not think about it) whilst the remaining participant said it was okay.

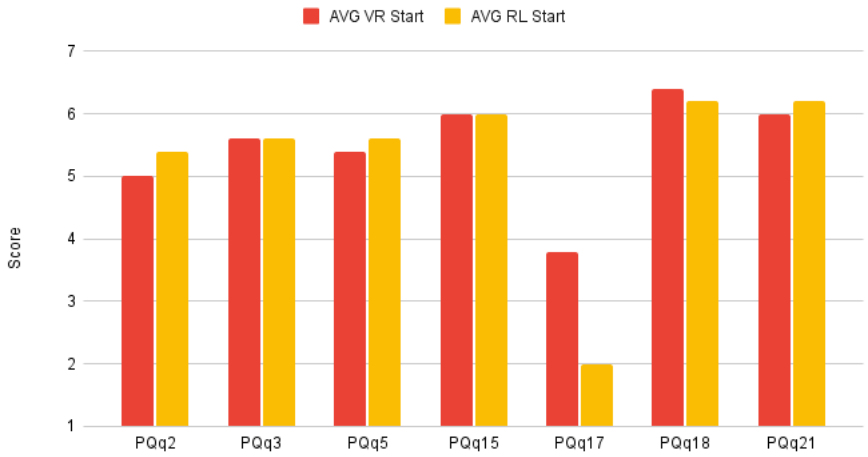

Figure 4.12: The results of the questions in the Presence Questionnaire that were in some way related to hand tracking. The questions were scored from 1 to 7 where 7 usually relates to higher presence and 1 no presence. PQq17 however is a negative question and thus that question's scale is reversed.

In figure 4.13 the results from our presence evaluation alongside each participant's PQ score are presented. There it can be concluded that eight out of ten participants had an above medium or higher presence (5 and above). The presence score and the presence evaluation were computed in the same way as in iteration 1.

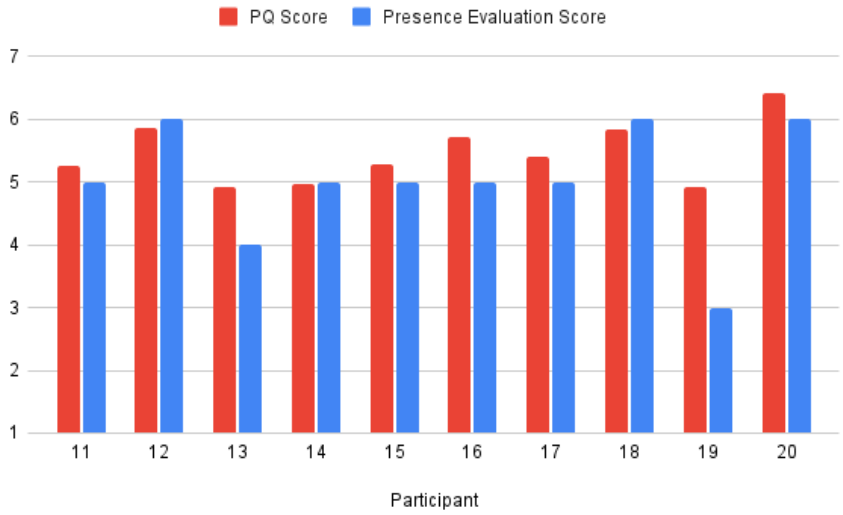

Figure 4.13: Each participant's PQ score alongside our presence evaluation of the same participant. Note that a discrete scale was used for the presence evaluation.

The VR experience seemed to have been appreciated by most participants. Eight out of ten participants said that the experience of doing the test was more interesting and felt more correct in VR. Though out of those eight, five stated that there was no difference regarding ARC. It was the same screen and the same application.

On the PQ question "How well could you localise sound?" the participants answered an average of 3.7 out of 7 where one meant "not at all" and seven meant "completely". The average score being in the middle of the scale hints towards uncertainty in the participants' perception. This can be further described by looking at their answers to the open discussion question during the VR experience. There were two distinct types of sounds the participants should be able to identify. The whirring of the engines and the ambient sound of other passengers. 6/10 said that the passengers talking were behind them, the other four did not specify a direction for the talking explicitly. The sound of the engines however was a more varied topic. In total, there were five different answers. two said the sound came from above them, two said it came from outside the window to the right, one said it came from behind them, two did not say a location and lastly, three participants said that they could not point to a specific location but that it came from everywhere just like in real aeroplanes.

When asked which of the three passenger alternatives (robots, dummies, or empty) they preferred, eight out of ten participants preferred the robot passengers. Four out of those eight said that the robots were the best available choice but did not necessarily like them.

All participants expressed some feeling of actually being in an aeroplane. Eight out of ten participants said that it was quite realistic or that they truly felt that they were on an aeroplane, while one said that the feeling was not that significant, just simply more than without VR. When asked what contributed the most to the feeling of realism seven out of ten participants mentioned the sound and all ten participants mentioned some visual aspects. There were however six distinct different answers:

- The sound and the environment as a whole (except the passengers): 4 participants.
- The sound and that you could see out through the window: 2 participants
- The sound combined with how crowded it felt: 1 participant.
- The visuals representation: 1 participant.
- To be able to look out of the window and see the clouds move: 1 participant.
- How crowded the cabin felt: 1 participant.

A summary regarding the participants' general opinions regarding the test as a whole (attitude), regarding hand tracking, and their presence, can be seen in figure 4.14.

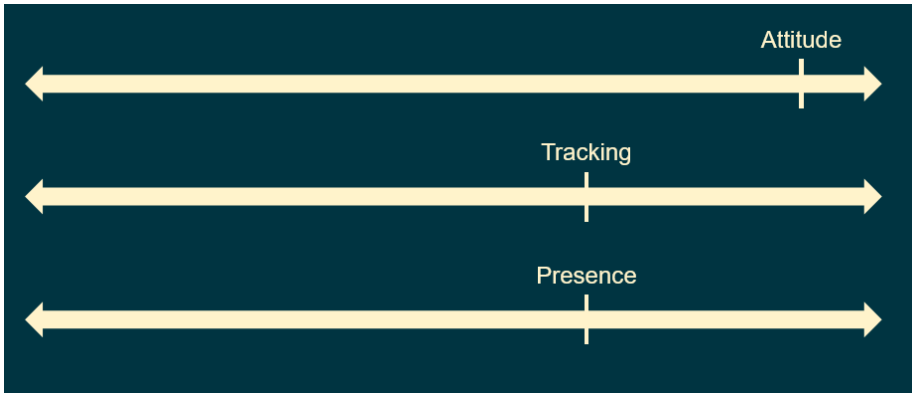

Figure 4.14: A summary of the participants' opinions regarding a few key aspects in regards to the tests done in VR in iteration 2. Right is more positive and left is more negative.

Is the VR controlling scheme intuitive and easy enough to use so as to not be a hindrance for testing, and to what extent?

All participants tried to touch the screen with their fingers without us having to explicitly explain that they had to. We also did not explain that the VR screen was the real screen this time.

There were still instances where hand tracking inaccuracies caused participants to miss buttons, however, it did not seem to have a significant effect. As stated before, no participant found the inaccuracies frustrating or irritating and some participants experienced no accuracy-related drawbacks. Additionally, the difference in average completion time for all VR tasks versus all RL tasks only differed by 4.8 seconds.

The touch indicator implemented to aid when a participant missed a button did not seem to hinder testing. Nine out of ten participants noticed the dot in VR out of which only two noticed it in RL. Participant 16 who only ever missed one button once did not notice the dot at all and participant 15 who experienced no accuracy-related drawbacks simply noticed that it appeared when they touched the screen in VR. Out of the remaining 8, seven used the indicator to figure out where they needed to press while one is unclear.

4.3.4 Result Analysis

From the task completion times presented in table 4.1 and figure 4.10 we can see that there is only a small difference between the time taken in VR versus the time taken in RL. Two things are worth noting. During participant 11's test in VR the map application encountered two bugs that disrupted the test, prolonging their test completion time without it being their or VR's fault. In hindsight that test should probably have been discarded because of this, but their opinion and answer to the post-test questions as an external participant (not a Tactel employee) still felt valuable. The other noteworthy thing was that participant 16 was during both scenarios very slow and careful with the test. Increasing the time taken for both VR and RL, with RL going to the point of reaching max time. It is probably due to this that they

encountered so few drawbacks. But the long test times do skew the result here, making them an outlier.

Over all, the task completion times were a lot more equal between VR and RL compared to the result in iteration 1. This relates to the number of drawbacks experienced which was also greatly reduced (see figure 4.11). The average amount of drawbacks seen during a test in iteration 1 was six drawbacks which is equal to the highest amount of drawbacks found for a participant in iteration 2, which had an average of 3.4 drawbacks found. Both instances of accuracy-related drawbacks and instances of readability ones were reduced.

For accuracy, the improvements to screen alignment as well as the increase in screen size made them less frequent. This resulted in the average amount of instances of accuracy-related drawbacks per test being reduced from 4.7 in iteration 1 to 2.9 in iteration 2. If they did miss, however, the new touch indicator helped the participant to quickly and efficiently figure out what went wrong and adjust where they needed to tap. This decreased the amount of time each error took to solve, which is reflected in the task completion times (see figure 4.10). Also worth noting is the improvement to the total amount of tasks completed with no drawbacks. In iteration 1 there were seven such instances, all occurring in task 2. In iteration 2 there were a total of 12, spread across all tasks. The low amount of drawbacks as well as the similarity between VR and RL task times indicates that VR is not as much of a hindrance to task completion as it seemed in iteration 1.

The increase in screen size also helped to increase the readability of text. Which in turn made the average amount of readability-related drawbacks per test decrease from 1.3 to 0.4 between the iterations. Speaking of view-based drawbacks. The reason for the screen blinking, as observed during the test, was never solved. Our theory is that since it occurred whenever the ARC application was not the active application on the computer, it might be that the Android emulator was too heavy for the PC, as it also ran the VR application and OBS on it at the same time. On the bright side, there were no indications that the VR application itself was too heavy for the computer as the bug could not be replicated regardless of how many other applications we ran at the same time except the Android emulator. Meaning it should not be counted as a negative for doing UX tests with VR.

The participants seemed to be more positive to their virtual hands in this iteration. Many answered that their hands in VR felt enough like their own hands so as to not think about them twice. Looking at figure 4.12, both groups of participants seemed to be above medium for all questions related to hand tracking (note that PQq17, "*How much did the hand tracking controlling scheme interfere with the performance of assigned tasks or with other activities?*", is a negative question and wants a low score). Additionally, for all questions except PQq17, there was no difference between starting in VR and in RL. For PQq17 it seems however that those who started in VR found that hand tracking interfered more than those who started in RL. This is especially surprising since the opposite was true in iteration 1. Looking at what each participant answered to PQq17 it is however clear that only two participants deviated from the remaining eight. These were participants 17 and 19, both part of the group that started in VR. They answered PQq17 with a five and six respectively. In the case of participant 17, until task three in VR, the hand tracking had been working very well. However, in task three

the hand tracking seems to have lost its accuracy. The participant was suddenly very off for two buttons, each requiring multiple presses. Since it was the last thing that occurred and since it was unexpected due to the rest working fine this could explain why they said the hand tracking interfered more than most other participants. Participant 19's reason on the other hand is more unclear. During the test only a few small misses occurred and there was no clear indication that the participant found the hand tracking limiting during the post-test interview. Yet they answered that the hand tracking interfered more than any other participant did in PQq17. Except for participant 17 and 19, all participants answered PQq17 with a three or less, where one meant *not at all* and 7 meant *interfered greatly*.

From iteration 1 we could clearly see that the inaccuracies of hand tracking could lead to frustration and irritation. No such feelings were reported in iteration 2. This was most likely due to both the improved screen alignment and screen size limiting the effect of hand tracking inaccuracy and the touch indicator helping the participant to understand and adjust to their misses. This in turn allowed the participants to focus more on the environment and test as a whole. These improvements are believed to be one of the main factors of the increase in both the PQ score and the presence evaluation score.

From the answers it is clear that multiple factors play a role in making the experience feel realistic. This is supported by the spread of answers to the question "What contributed the most?". Very few of the answers could be grouped as we can see in the result. Only the answer "Both the visual and sound combined" was given by multiple participants, and even then, which visual differed between the answers. Otherwise, each participant had their own combination or single reason for what was most impactful. To us, this shows a clear consensus even though the answers are spread. There is not one thing that contributes the most to getting a realistic experience but rather all parts equally working together. This of course means that all parts need effort and quality put into them to reach the level of realism wanted. This aligns with what was previously found by Potter et al. [19].

Our hypothesis was that exchanging the dummies for robots would have a positive effect on the environment since making them clearly not human would remove the uncanny valley effect. This was done even though most participants from iteration 1 mentioned that they wanted more realistic passengers. The result regarding what passenger alternative the participants preferred (the robots) as well as their general comments regarding those supported our hypothesis. Additionally, most participants reacted strongly negatively when presented with the dummies in this iteration. It was not just that they preferred the robots it was that they were actively against the dummies. Which was interesting, when participants could compare different sets of passengers then they were much more negative towards the dummies compared to when they were the only choice shown as in iteration 1. Which was strong evidence for our theory. One should however note that we did not test with more realistic passengers and as such do not know what effect that would have had on the simulation.

From the participants' answers to where the sound came from it is clear that they could not localise it. Not very surprising considering the fact that the sound did not have any direction and indeed was only two-dimensional. However, all participants either believed that they could point to where the sound came from or justified that the sound comes from every-

where on an aeroplane meaning it was hard to localise. Especially interesting is that six out of ten participants all said the sound of people talking came from behind them. Many justified this with the fact that they could only see robots in front of them and to the side of them, and thus the people had to have been behind them. Had they turned around they would however only have seen more robots. Participant 19, who did actually turn around, realised that the sound moved with how they turned and thus stated that the talking always came from behind them. The fact that our scenario was in an aeroplane probably helped make our unrealistic sound seem more realistic. Had the setting been in a place where it usually is easier to localise sound it might not have had the same positive effect on presence. This could be further supported by participant 19 having the lowest presence score for iteration 2, though other factors of course also play a large role in the presence evaluations.

One concern about the result is that our selection of participants was very narrow. As mentioned, due to administrative difficulties the test was mainly conducted with employees from the Tactel office, with some addition of the few family and friends that had some spare time during work hours. Even though we constructed a new test from scratch it became obvious during testing that the majority either had worked with the application, or more commonly, seen the application before and participated in other usability tests. Meaning that many already knew exactly the steps needed to complete the tasks beforehand. This conformed with our hypothesis that they would be fast. However, the average time between employees and external participants did not differ as much as we first believed. As we only took in external in iteration 2 we can only compare from that iteration. Participants 11, 13, 14, and 18 were fully external, while participants 17 and 19 were thesis workers who had slightly more knowledge about the product but were still not on the same level as full employees. And from the figure 4.10 one can see that there were both faster and slower employees. Indicating that a participant's background knowledge did not matter as much because of the way the test was constructed.

Chapter 5
Discussion

In this chapter important and interesting parts of the thesis will be discussed. First, the research questions asked in the beginning of this thesis will be answered. Secondly, we go through some aspects of the thesis deemed important but did not fit in the result analyses. Lastly, we will look into possible future work this thesis can lead to.

5.1 Research Questions

What are the limitations and drawbacks of the current method for usability testing when a natural setting is not feasible?

From our literature study as well as our continuous discussion with Tactel's design team we have come to the following conclusion. No matter how extensive the research, usability testing, heuristic evaluations, and other assessments may be, it becomes highly probable that the product will elicit different reactions when used by actual consumers if the testing environment fails to replicate the natural setting.

As an example: In this thesis, we have studied a scenario that not necessarily needed the natural environment but where the test could be improved by having it. But as Kinateder et al. [9] studied with fire safety, sometimes the natural setting cannot be reproduced due to the danger it could put the participants in but is needed to get a correct reaction. The argument that the natural setting lets users behave as they would stay the same. But now when the alternative is to set up huge expensive tests where every aspect needs to be under control then the cost also becomes a limiting factor. It is possible to set up a scenario mimicking a real fire hazard but the cost of it makes it not feasible and not having either real or fake fire limits how valid the test results are. In our case, it is only costs, both monetary and time, that prevent the usability tests to be conducted in the product's intended environment.

What is gained and what is lost when using Virtual Reality as a medium for usability testing compared to normal usability testing?

One thing multiple participants commented on was the effect of not being able to see the moderator. Those who were positive to not seeing the moderator all commented on feeling more free and able to explore and investigate the application than when they could see the moderator. This aligns with previous findings, that we act differently when we feel that we are observed and tested [2]. Furthermore, when moderating it is important to try to remain neutral, which can be difficult when you need to keep track of both your voice and body language [12]. This is then simplified with VR since the moderator is not visible when which means the body language is no longer a factor. On the other hand, some participants commented that it felt weird to talk to someone that they could not see, making them less comfortable asking questions while in VR.

Another factor is the environment that VR brings. Testing in a natural setting better shows what the participants would actually do with the product. From our testing, we saw that the participants did have compelling feelings of being on an actual aeroplane. That is, they did feel like they were in the product's intended environment. Furthermore, VR also brings the possibility to easily change the environment. In our case, we simply used this to explore what the participants preferred in terms of other passengers. With further development it is however not difficult to, for example, change the time of day, aircraft model, etc, during a test or between different participants.

VR also allows the observer to see what the participant sees. In our case, the application ran on Windows making the participant's view visible by default. This could then be used as a first step to track what the participants are looking at. Though not implemented in this thesis, Meta Quest Pro is able to use eye-tracking, which could further enhance the ability to see exactly what a participant is focusing on. Furthermore, VR would also allow the user's perspective to be recorded in combination with eye-tracking to both see what they notice and what is in their field of view. It would also enable the researchers to see things such as if the participant's hand hides a button when pressing another. Eye-tracking and perspective recording can technically be done without an HMD but with a Meta Quest Pro, you get them in addition to VR with no added tools.

Though we did not see the effects of it, VR could in some cases deter some users, or for other reasons limit who can participate. In our case, all participants were adults either still studying or working, and most had at least some interest in VR. If the tests would have been conducted with children or the elderly then VR might have been more difficult to use. In these cases, one would most likely need to aid the participants more with the HMD which could increase workload and thus decrease the efficiency during testing. Additionally, people who know that they get cybersick might prefer to not take part at all. Out of the 20 total participants, two described mild symptoms of cybersickness, though not until after taking off the HMD. It probably helped that the participants were stationary during the test and nothing in the surrounding environment was moving fast. Introducing more movement, both in the form of things in the cabin moving, but also the participants themselves moving, might be an interesting addition for future research to further increase realism. Which then would

force the researchers to take potential cybersickness more into account.

How does Virtual Reality affect usability testing when natural settings otherwise cannot be reconstructed in regards to

Cost:
As our VR testing extends the normal lab-based usability testing, we consider the existing test costs as base costs and solely concentrate on the additional costs associated with VR. We can split costs into two categories. Continuous and one-time costs. One-time costs may differ from project to project so we can not predict every scenario. What we can do however is give an account of what this study would have costed if we had started from scratch and everything had needed to be bought. Following is a list of hardware used in this study and an approximation of their cost (including taxes) as of the day of writing (2023-05-10).

- Computer: ~35 000 sek
- Meta Quest Pro kit: ~14 000 sek
- Link Cable: ~1 000 sek
- Touch screen 17": ~5 000 sek

Bringing it to a total of ~55 000 Swedish crowns. Which is a low to medium-sized cost for an established IT company. One can probably get away with a similar or adequate setup for cheaper depending on the hardware chosen. For continuous costs on the other hand we can not find any more than the additional salary for developers to create and maintain the simulation program.

There is also the question of non-monetary costs such as development time. This prototype we have created here, albeit highly functional is still just that, a prototype. Creating a fully functional well-made simulation could take months more work. Also if the program needs to change for each new test then it becomes a very time-consuming task. However, if instead the program is made modular as we have tried with ours, such that you can plug any screen into it and run different programs on it, then the time cost can be seen as a one-time cost with maybe some additions and patches down the line. Lowering the amount of development time drastically.

Cost is difficult to determine what is worth and what is not. The feedback we got from Tactel is that they consider the money spent worth the prospect of VR development. Also that the products bought now can be used for other projects if they decide that VR was not worth it. Meaning that the one-time costs can often be re-used and are therefore less of an impact economically.

To summarise: VR is definitely more expensive and time-consuming than normal usability testing. However, if it is too expensive is a subjective matter highly dependent on the organisation's economic status as well as the project's complexity. Meaning that we can not say whether the cost is worth what you gain and instead leave that as a decision for the reader.

Efficiency:
After the fixes made in the second iteration we could see that the time needed to complete tasks as well as the number of errors caused by VR both went down. The time it took for participants to complete their tasks in VR approached the real-life ones but was still slower, as seen in figure 4.10. The similar times indicate that VR does not make participants much slower and therefore does not pose much of a hindrance. The number of VR-related drawbacks was significantly fewer in iteration 2 but no test was completely fault free. Showing that VR does impose problems that could take the participant's focus off the real task and disturb the test. Most of the time when an instance of faults happened they were solved quickly, not taking much time or cognitive effort for the participant. These two sets of data show that the efficiency when testing in VR is close to the same in real life but still lower.

The test we created for this was a rather simple and straightforward one. With only three simple tasks we never tested how the participants would behave when challenged while also having to deal with VR. Therefore it is hard to say if VR and RL testing are as close as the numbers make them seem or if the test just was too simple for the differences to show.

The problem where the resolution rendered certain elements to be unreadable, is one of the main reasons we can see why VR might not be usable in testing in terms of efficiency. Our first iteration showed that users struggled with elements like reading smaller text when a 12" screen was used. This shows that some tests will be impossible to perform in VR correctly, if they need a similar or smaller screen or if the application itself uses smaller elements. The errors we have seen will then show up more, disturb the participant and take time and cognitive effort from them, decreasing efficiency. Hopefully, more testing with different headsets and setups can show a solution to this problem. Otherwise the decision of using VR or not will be up to the test researchers when constructing the test, asking themselves if they can use VR with the product they test. Speaking of testers, this leads us to the aspect of efficiency for them. Using VR introduces more steps to testing in terms of setup and things to control during the test. For our project most things can be set up once before the first test and be kept running for the rest of the testing day, meaning a minimal increase in workload. Provided one has multiple tests a day. But we can see how testing as a whole can go down in efficiency if the VR application needs to be reset between each test.

To summarise: Overall we saw a small decrease in efficiency in VR. The main culprits being tracking-related problems and resolution problems. If constructed with this in mind the test can be run closely similar to that of real life. Heavily depending on the complexity of the product tested but we can definitely see this being used for usability testing in terms of efficiency in the correct circumstances.

Perceived product experience:
From our tests we can determine that the participants' opinions about the IFE application tested, Arc maps, did not change much. The consensus seemed to be that it was the same regardless of environment, showing that the perceived product experience did not decrease due to VR, but also that it did not increase. Experience of the test as a whole on the other hand was drastically improved. The consensus was that it felt more like they actually were on their way to vacation. For some, the environment engaged their senses more and made them

interested in playing around with the product, testing and using it as they would normally on a plane. Some participants felt less like they were being tested and more like they were just using the product. Aspects such as not seeing the moderator or hearing the typing of the observer seemed to decrease the feeling of being tested for some participants, which in turn increases their overall experience as well as the validity of their answers.

Subjective workload:
During testing no instances where VR created any problems like headaches or cybersickness were noted. The closest any participant came to feeling unwell due to VR was participant 19 and 20 who both expressed some strain on the eyes as well as mild dizziness after taking off the headset. Regarding this question, however, we know that our usability test was not created to test this properly. Our scenario was a short and calm one where the participant sat down the whole time and nothing moved around wildly in the scene around them. Nothing that normally induces cybersickness. We can therefore only with confidence say that participants' subjective workload is not increased by being in VR as long as the test is on the calmer side. To more accurately answer the question more different tests are needed.

Validity:
We can see two main ways VR impacts the validity of the tests. We have already gone in-depth on both of these topics and will therefore keep it short. On one hand, it decreases the validity due to the extra drawbacks in the form of blurriness and inaccuracy created by VR. On the other hand, it increases the validity because the participants felt less tested and more exploratory instead, due to not seeing the testers and being in a more natural environment.

Also worth noting when talking about validity is that, how we have tested VR fits only certain ways of conducting usability tests. The way our tests were constructed works well for the "normal" lab setup. While the more exploratory testing styles where the moderator is more involved or there is more than one participant at a time do not work in this format.

5.2 Problems and points of interest

5.2.1 Participants' answers and opinions

While analysing the result during the first iteration we came across some participants whose answers to the debriefing interview, the presence questionnaire, or our observations of what happened during the test did not fit together. Sometimes the opinion shared with us seemed more positive than expected. We know this to be a often occurring phenomena in testing especially when recruiting internally at the company. Participants could either be just hyped about the new technology developed at the place they are working, or too afraid to hurt us by criticising our work. They could also just have genuinely liked the product and simulation, we cannot be sure. Regardless, some more positively weighted answers could be accounted for when analysing the results.

The other group of problematic answers however posed more of a problem. Some participants gave contradictory answers. Either due to them changing their opinion when thinking

about it alone, being more positive when talking to the moderator, not wanting to criticise our work in front of us, or due to them not fully understanding the questions in the PQ or the post-test interview. For example, they could state that they only slightly felt like they were on an aeroplane during the post-test interview but then answer in the PQ that they had a very strong feeling of being on an aeroplane moving through the air. This of course lowered the validity of our result and result analysis as we cannot say for sure how they actually felt about using the tested product in the simulation created.

Both these problems could potentially have been solved (or at least diminished) by using the "devils advocate" interviewing technique mentioned at the beginning of the report. By taking a stance opposite to the participant one could have coaxed out their true feelings if they were shown that it was "allowed" to criticise the product. We could also test how strongly they felt about their answer by confronting it. Which would have been especially useful in the cases where we felt the participant contradicted themselves. In the end, however, we opted to not use this technique as we felt we were too green in the role of moderator to correctly use it. We were afraid of running into the problems that Rubin and Chisnell talked about [12], that we would not improve the validity of the answer as much as just bias it.

5.2.2 Tracking

One of the biggest problems found during development was tracking. For our concept to function properly it was very important that the user's virtual hands' positions relative to the virtual screen were identical or at least close to the user's real hands' positions relative to the real screen. Unfortunately, neither the hands nor the screen could be tracked perfectly. For example, the position of the hands in VR would shift depending on the angle that the HMD viewed them from. The changes in iteration 2 did improve the tracking to an extent, making VR closer to equivalent to RL for task completion, but it did not fully eliminate all errors.

For this thesis we used a Meta Quest Pro as that was the only HMD we had access to. Meta Quest Pro has the limitation of not being able to use real-life trackers to define real-world objects but other HMDs do, for example, Varjo [48]. Placing such trackers at the edges of the real screen could have greatly helped with aligning the virtual screen to the real. Spatial anchors are good but they do drift slightly over time and are almost never fully consistent between frames. Additionally, if the HMD loses tracking and needs to redefine its surroundings then the spatial anchors placed will most surely also be lost, while real-world trackers would stay in the same position.

5.2.3 Using a bigger screen

Another limitation came in the form of the screen being used. Since we wanted the VR screen to be mapped one-to-one with the real-world screen then they also had to be the same size. However, once in VR, the resolution of the screen is not only limited to the real screen's resolution but also to how many pixels it covers in the HMD. In other words, the resolution of the screen is also affected by how much of the field of view in the HMD that it takes up. Unfortunately, the usual economic seating IFE screen is not very big. This made it hard and draining, bordering in some cases impossible, to try to read smaller fonts on the screen. As

we explored in the second iteration, using a bigger screen did impact the resolution of it in VR. Going from 12 to 17 inches made most text readable for all participants. There were some who still struggled somewhat and needed to lean in to see but the overall improvement was notable. However, this can not be viewed as a feasible solution to the problem. For one, the screen size is not "correct" in comparison to a real IFE screen. Meaning that if we wanted to construct a simulation that should mimic reality then we already have a mismatch. It also means that if we ever wanted to test an even smaller screen, like a mobile phone which we talked about in section 3.2.3 but never tested, then it could be impossible to read anything on it due to its small size.

5.3 Future work

A first next step towards improving our program would be to add eye-tracking functionality. As mentioned before, eye-tracking would be a tremendous addition to usability testing as testers would get raw objective data over what the participants actually notice and look at. As mentioned in section 2.2.3, eye-tracking is not a new concept, not even in the context of usability research. It is however a great benefit with VR that eye-tracking can be added on top of what has already been developed without the need for any additional devices.

Another area of improvement would be creating different scenarios and environments. This study was limited to only one of each, so exploring VR's strength with being able to switch testing-environment could be interesting. For example, what if the user was in a very stressful situation, or what if something else was constantly trying to distract them? In our environment, the participant sat only in the economy class and nothing but them moved in the cabin. Multiple participants mentioned that they wanted more movement in the aeroplane to make it feel more real. We did not have the time to fully implement animations, though it could have had a great impact on the results and the participants' presence.

This study was limited both in terms of hardware used as well as the duration of each test. For example, the Meta Quest Pro's strength lies in its ability to stand on its own legs. It is light, great at running standalone applications, and contains multiple experimental features like passthrough and eye-tracking. But it is not necessarily the best for usability testing in this manner. It would thus be interesting to test using different HMDs that have other strengths and weaknesses, for example Varjo as previously mentioned. Being able to place real-world markers would allow easier setup for the testers and could eliminate the drifting and inaccuracies of the anchors. Then there was the other limiting factor that our tasks were short and relatively simple. As they needed to be done twice and the whole test process needed to stay under an hour we could not make them more in-depth. So there is plenty of room to test longer and more complex usability tests and see how the drawbacks of VR impact them.

Broadening the group of participants could, as stated before, also affect the result, thus being an interesting aspect to look into in future testing. Our participants had a high collective technical knowledge and experience and almost all had some amount of prior VR experience. Testing other groups of people would enhance our findings tremendously and is needed to better establish whether VR truly can be used as a medium for usability testing.

Chapter 6
Conclusions

Is VR viable as a medium for usability tests? We believe so yes. Although there still are some problems with tracking and resolution, we believe that the pros of having participants in a natural setting where you otherwise cannot outweigh the negative aspects of VR. Especially since we, two students with very limited time could create a functional program that showed only a small difference in results between virtual reality and real life. What then could a company or institution make with a greater budget, time, and expertise?

In regards to the different tested aspects of VR as a medium for usability testing we can say that VR:

- increases costs,
- somewhat decreases efficiency,
- improves the experience of the test while perceived product experience remains unchanged,
- increases validity,
- does not increase subjective workload for relatively simple tests,
- may limit which testing setups and participants that can be used,
- does not impose much additional workload for testers.

It is, of course, important to note the limitations and simplicity in our study, and that we have only tested one way of conducting usability tests. There is much further work and research needed before one can truly say whether VR can be used as a medium for usability testing or not, but we do believe that it is something worth the effort.

Bibliography

[1] Helen Sharp, Yvonne Rogers, and Jennifer Preece. *Interaction Design: beyond human-computer interaction*. Wiley, 5 edition, 2019.

[2] Juergen Sauer, Andreas Sonderegger, Klaus Heyden, Jasmin Biller, Julia Klotz, and Andreas Uebelbacher. Extra-laboratorial usability tests: An empirical comparison of remote and classical field testing with lab testing. *Applied Ergonomics*, 74:85–96, 2019.

[3] Ivan Blagojević. 99Firms, Virtual Reality Statistics, Accessed: 2023-01-26. Available from: https://99firms.com/blog/virtual-reality-statistics/#gref.

[4] Tactel AB. Tactel, Företaget, Människorna och vår värld, Accessed: 2023-01-26. Available from: https://tactel.se/sv/om-oss/.

[5] United Nations. THE 17 GOALS, Accessed: 2023-06-11. Available from: https://sdgs.un.org/goals.

[6] United Nations. Goal 9, Accessed: 2023-06-11. Available from: https://sdgs.un.org/goals/goal9.

[7] United Nations. Goal 13, Accessed: 2023-06-11. Available from: https://sdgs.un.org/goals/goal13.

[8] Fabio Freitas, Henrique Oliveira, Ingrid Winkler, and Marcus Gomes. Virtual reality on product usability testing: A systematic literature review. In *2020 22nd Symposium on Virtual and Augmented Reality (SVR)*, pages 67–73, 2020.

[9] Max Kinateder, Enrico Ronchi, Daniel Nilsson, Margrethe Kobes, Mathias Müller, Paul Pauli, and Andreas Mühlberger. Virtual reality for fire evacuation research. In *2014 Federated Conference on Computer Science and Information Systems*, pages 313–321, 2014.

[10] Effi Freya Picka, Annika Vogel, Marie-Sophie Roder, Jonas Birkle, Daniela Schrenk, Jessica Linnemann, Julia Moritz, and Stefan Pfeffer. Virtual usability testing (virtuse) - development of a methodical approach for usability testing in vr. In Constantine

Stephanidis, Margherita Antona, and Stavroula Ntoa, editors, *HCI International 2022 Posters*, pages 109–116, Cham, 2022. Springer International Publishing.

[11] International Organization for Standardization. ISO 9241-11:2018(en) Ergonomics of human-system interaction — Part 11: Usability: Definitions and concepts, Accessed: 2023-01-30. Available from: https://www.iso.org/obp/ui/#iso:std:iso:9241:-11:ed-2:v1:en.

[12] Jeffrey Rubin and Dana Chisnell. *Handbook of Usability Testing*. John Wiley and sons ltd, 2 edition, 2008.

[13] Don Norman. *The Design of Everyday Things: Revised and Expanded Edition*. Basic Books, 2013.

[14] Michael Heim. *Virtual realism.* Oxford University Press, 1998.

[15] Paul Milgram, Haruo Takemura, Akira Utsumi, and Fumio Kishino. Augmented reality: a class of displays on the reality-virtuality continuum. In Hari Das, editor, *Telemanipulator and Telepresence Technologies*, volume 2351, pages 282–292. International Society for Optics and Photonics, SPIE, 1995.

[16] Philipp A. Rauschnabel, Reto Felix, Chris Hinsch, Hamza Shahab, and Florian Alt. What is xr? towards a framework for augmented and virtual reality. *Computers in Human Behavior*, 133:107289, 2022.

[17] Márcio C. F. Macedo and Antônio L. Apolinário. Occlusion handling in augmented reality: Past, present and future. *IEEE Transactions on Visualization and Computer Graphics*, 29(2):1590–1609, 2023.

[18] Mel Slater. Measuring Presence: A Response to the Witmer and Singer Presence Questionnaire. *Presence: Teleoperators and Virtual Environments*, 8(5):560–565, 10 1999.

[19] Thomas Potter, Zoran Cvetković, and Enzo De Sena. On the relative importance of visual and spatial audio rendering on vr immersion. *Frontiers in Signal Processing*, 2, 2022.

[20] Bob G. Witmer and Michael J. Singer. Measuring Presence in Virtual Environments: A Presence Questionnaire. *Presence: Teleoperators and Virtual Environments*, 7(3):225–240, 06 1998.

[21] Bob G. Witmer, Christian J. Jerome, and Michael J. Singer. The factor structure of the presence questionnaire. *Presence: Teleoperators and Virtual Environments*, 14(3):298–312, 2005.

[22] Mel Slater. How Colorful Was Your Day? Why Questionnaires Cannot Assess Presence in Virtual Environments. *Presence: Teleoperators and Virtual Environments*, 13(4):484–493, 08 2004.

[23] Takashi Shibata. Head mounted display. *Displays*, 23(1):57–64, 2002.

[24] Meta Quest. Set Up Hand Tracking, Accessed: 2023-01-27. Available from: https://developer.oculus.com/documentation/unity/unity-handtracking/.

[25] Google VR. Degrees of freedom, Accessed: 2023-06-09. Available from: https://developers.google.com/vr/discover/degrees-of-freedom.

[26] Lisa Graham, Julia Das, Jason Moore, Alan Godfrey, and Samuel Stuart. The eyes as a window to the brain and mind. In Samuel Stuart, editor, *Eye Tracking, background, methods and applications*, chapter 1, pages 1–15. Humana Press, 2022.

[27] Humanities Lab Lund University. Eye Tracking, Accessed: 2023-05-09. Available from: https://www.humlab.lu.se/facilities/eye-tracking/.

[28] Meta. Eye Tracking Privacy Notice, November 2022. Available from: https://www.meta.com/en-gb/help/quest/articles/accounts/privacy-information-and-settings/eye-tracking-privacy-notice/.

[29] Ji-Un Hwang, Ji-Seon Bang, and Seong-Whan Lee. Classification of motion sickness levels using multimodal biosignals in real driving conditions. In *2022 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 1304–1309, 2022.

[30] Lisa Rebenitsch and Charles Owen. Review on cybersickness in applications and visual displays. *Virtual Reality*, 20(2):101–125, Jun 2016.

[31] Unity. Creating games for Android, Accessed: 2023-01-25. Available from: https://unity.com/solutions/mobile/android-game-development.

[32] Unity. Build for virtual reality with Unity, Accessed: 2023-01-26. Available from: https://unity.com/unity/features/ar.

[33] Unity. Augmented reality, Accessed: 2023-01-26. Available from: https://unity.com/solutions/vr.

[34] Unity. Shader Graph, Accessed: 2023-02-24. Available from: https://unity.com/features/shader-graph.

[35] Blender.org. About Blender, Accessed: 2023-02-23. Available from: https://www.blender.org/about/.

[36] Free Software Foundation Inc. GNU General Public License, version 2, 1991. Available from: https://www.gnu.org/licenses/old-licenses/gpl-2.0.html.

[37] OBS. OBS Studio, Accessed: 2023-03-02. Available from: https://obsproject.com/.

[38] The Apache Software Foundation. Apache License, Version 2.0, 2004. Available from: https://www.apache.org/licenses/LICENSE-2.0.

[39] Romain Vimont, Yu-Chen Lin, brunoais, yanfl, flying press, Harsh Shandilya, Sean, Alberto Pasqualetto, xeropresence, Tom Ripley, and Cccc_. scrcpy (v1.25), Accessed: 2023-02-23. Available from: https://github.com/Genymobile/scrcpy.

[40] Meta Quest. Spatial Anchors Overview, Accessed: 2023-02-23. Available from: https://developer.oculus.com/documentation/unity/unity-spatial-anchors-overview/.

[41] Pedro Monteiro, Hugo Coelho, Guilherme Gonçalves, Miguel Melo, and Maximino Bessa. Comparison of radial and panel menus in virtual reality. *IEEE Access*, 7:1–1, 08 2019.

[42] Jeremy Birn. 3dRenderer Glossary: Fresnel effect, 2001. Available from: http://www.3drender.com/glossary/fresneleffect.htm.

[43] Figma. Figma, the collaborative interface design tool, Accessed: 2023-03-10. Available from: https://www.figma.com/.

[44] Masahiro Mori, Karl F. MacDorman, and Norri Kageki. The uncanny valley [from the field]. *IEEE Robotics & Automation Magazine*, 19(2):98–100, 2012.

[45] Susan Farrel. Observer Guidelines for Usability Research, 2016. Available from: https://www.nngroup.com/articles/observer-guidelines/.

[46] Jacob Nielsen. Thinking Aloud: The #1 Usability Tool, 2012. Available from: https://www.nngroup.com/articles/thinking-aloud-the-1-usability-tool/.

[47] Microsoft collaborators. Microsoft PowerToys: Utilities to customize Windows, 04 2023. Available from: https://learn.microsoft.com/en-us/windows/powertoys/.

[48] Varjo. Varjo markers, Accessed: 2023-03-16. Available from: https://developer.varjo.com/docs/get-started/varjo-markers.

Appendices

Appendix A
Figures and Images

Table A.1: Part 1 of compilation of answer from the post-test interview in iteration 2, used for presence evaluation. Note that this is more of a mind map rather than fully compiled data.

Participant	PQ score	PQ score utan PQq9	roligare	accuracy upplevelse
11	5,25	5,391304348	x	dålig på små saker
12	5,875	5,913043478	intressantare	lite off, behövde trycka lite mer
13	4,916666667	5,043478261	intressantare	5mm off
14	4,958333333	4,956521739	intressantare	lite off ibland, ganska naturligt over all
15	5,291666667	5,347826087	samma	korrekt
16	5,708333333	5,782608696	häftigare, intresantare	ganska korrekt, bättre än förväntat
17	5,416666667	5,47826087	intressantare	dålig, bra i början men väldigt fel vid ett tillfälle
18	5,833333333	5,826086957	intressantare	tillräckligt korrekt, kanske fel någon gång
19	4,916666667	5	positiv	lite off, små saker svåra
20	6,416666667	6,391304348	intressantare	0.5 cm fel, kankse bara en känsla, väldigt nära iallafall
avg/consensus	5,458333333	5,513043478	intressantare	bra men ej perfekt

Table A.2: Part 2 of compilation of answer from the post-test interview in iteration 2, used for presence evaluation. Note that this is more of a mind map rather than fully compiled data.

Participant	fokus i VR	ARC VR/RL	Adaption technique	moderator skymd VR
11	x	samma	x	x
12	x	samma	korrigerar med pricken	x
13	x	Bättre i VR	korrigerar med pricken	nej
14	x	samma	korrigerar med pricken	drog ner presence
15	x	samma	x - behövdes ej	x
16	mer engagerad	samma	x - behövdes ej	x
17	x	samma	korrigerar med pricken	x
18	x	Bättre i VR	korrigerar med pricken	mindre styrd, fick göra mer vad hen ville
19	x	samma	korrigerar med pricken	konstigt när man vill ställa fråga
20	x	lite bättre i VR (Stolt)	korrigerar med pricken	x
avg/consensus	x	samma	korrigerar med pricken	x

Table A.3: Part 3 of compilation of answer from the post-test interview in iteration 2, used for presence evaluation. Note that this is more of a mind map rather than fully compiled data.

Participant	real scenario bild	flygkänsla	hands feeling
11	nej, i exp-rummet på en stol	ganska mycket (förutom passagerarna)	Tänkte ej så mycket på det, snabb
12	nej, på kontoret, testar vilken skärm som helst	kännde som att det var riktigt	Tänkte inte på det, så som vanligt förmodligen
13	ja, på flygplan, fokus mest på skärmen så enkelt att fantisera	kändes som flygplan, saknade rörelsen hos passagerarna	ok
14	nej, inte direkt	60-70% ganska övertygande	90% lite fel i vissa positioner
15	nej, ignorerar testomgivning (både i VR och RL)	Fick ändå rätt vibe	var bra
16	nej, inte alls	ganska trovärdigt, väldigt tydligt	ingen vänsterhand i början, annars bra
17	nej, vara bara här i rummet	ändå mycet, mer än förvänatat	var bra, märkte inte om de segade
18	nej	ganska trovärdigt	perfekt
19	nej, mer fokus på skärmen bara	bättre än utan, men passagerarna var för overkliga	tänkte ej på det, så förmodligen bra
20	nej, inte alls, vara bara här	hen satt på flyget	inte helt men nära hens faktiska händer
avg/consensus	nej	ganska trovärdigt	bra, nära naturligt

Table A.4: Part 4 of compilation of answer from the post-test interview in iteration 2, used for presence evaluation. Note that this is more of a mind map rather than fully compiled data.

Participant	Var kommer ljudet ifrån?	aditional notes	Presence Evaluation	Bästa passagerarna
11	uppifrån, utåt, ej inifrån		5	robotar
12	svårt att säga, som på ett vanligt flygplan		6	utan
13	ovanifrån, snett uppåt		4	robotar
14	pratar bakom. motor och luft från höger	att kunna se neråt sänkte presence lite i början	5	robotar
15	pratar bakom		5	robotar
16	bruset tystare än vanligt plan, prat bakom (ser dem inte)	vill utforska mer i VR, mer engagerad	5	robotar
17	brus från motorer, prat bakom		5	robotar
18	svårt att säga, förväntat då det är ett flygplan, prat bakom		6	robotar
19	brus alltid bakom oavset hur hen vänder sig, prat oklart	skärmen flimrade, störde	3	dummies
20	brus runtomkring, annars svårt	lite yr efter VR upplevelsen, ej innan eller under	6	robotar
avg/consensus	prat bakom, olika åsikt om motorljud (brus)		5	robotar

Appendix B
Questionnaires and Surveys

B.1 Informed Consent (in Swedish)

Samtyckesblankett

Samtycke till att delta i studien: *Evaluating Virtual Reality as a Medium for Usability Tests*

Jag har informerats muntligt om studien och samtycker till att delta.
Jag är medveten om att mitt deltagande är helt frivilligt och att jag kan avbryta mitt deltagande i studien utan att ange något skäl.
Min underskrift nedan betyder att jag väljer att delta i studien och godkänner att Lunds universitet behandlar mina personuppgifter i enlighet med gällande dataskyddslagstiftning och lämnad information.

☐ Jag väljer att delta i studien och godkänner att Lunds Universitet behandlar mina personuppgifter i enlighet med gällande dataskyddslagstiftning och lämnad information.

.....

Underskrift

....

Namnförtydligande Ort och datum

Studieansvariga:
Fredrik Voigt
Rasmus Andersson

Handledare:
Joakim Eriksson
Joel Jonsson

B.2 Screening survey

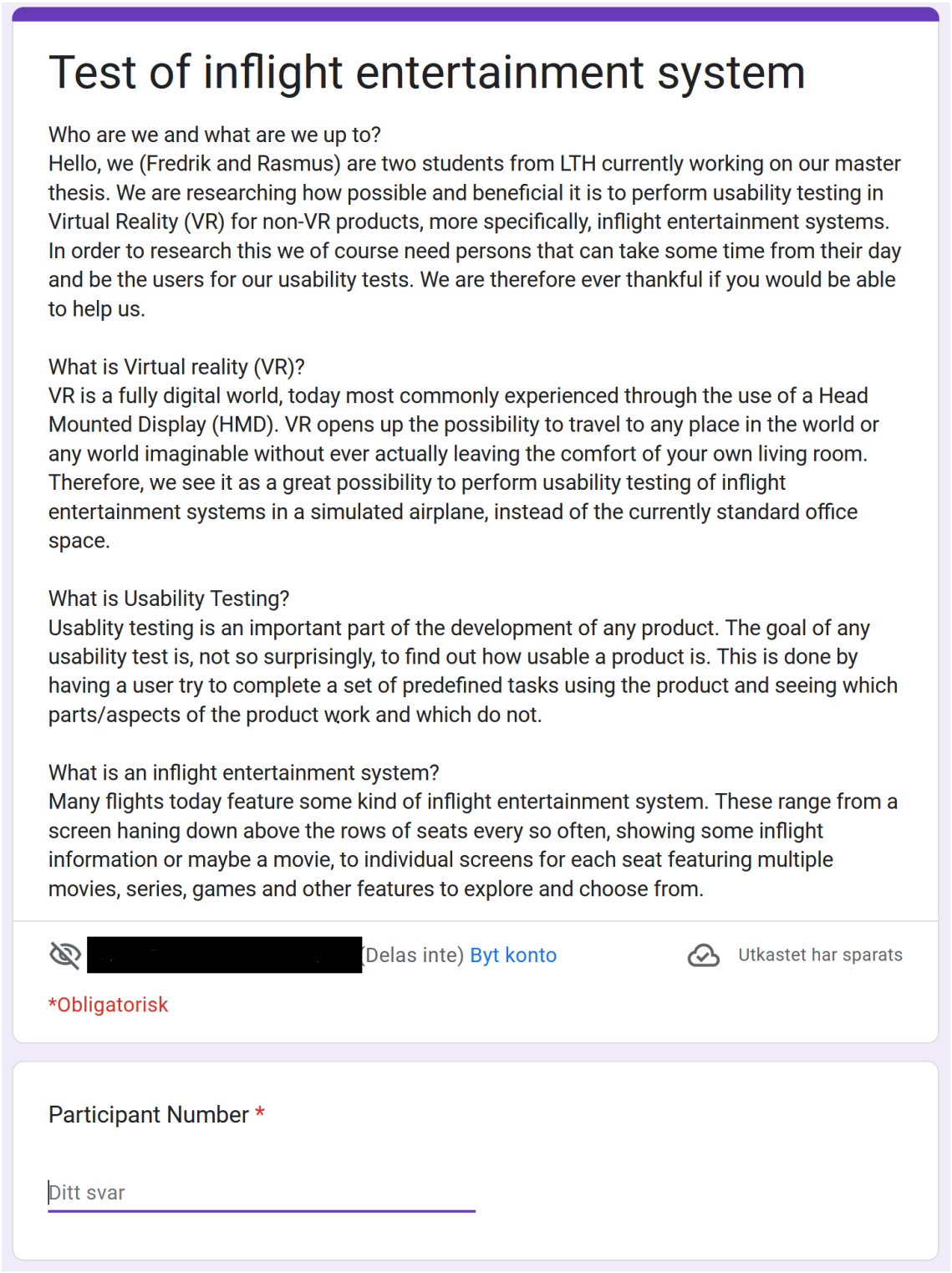

Figure B.1: Screening questionnaire section 1

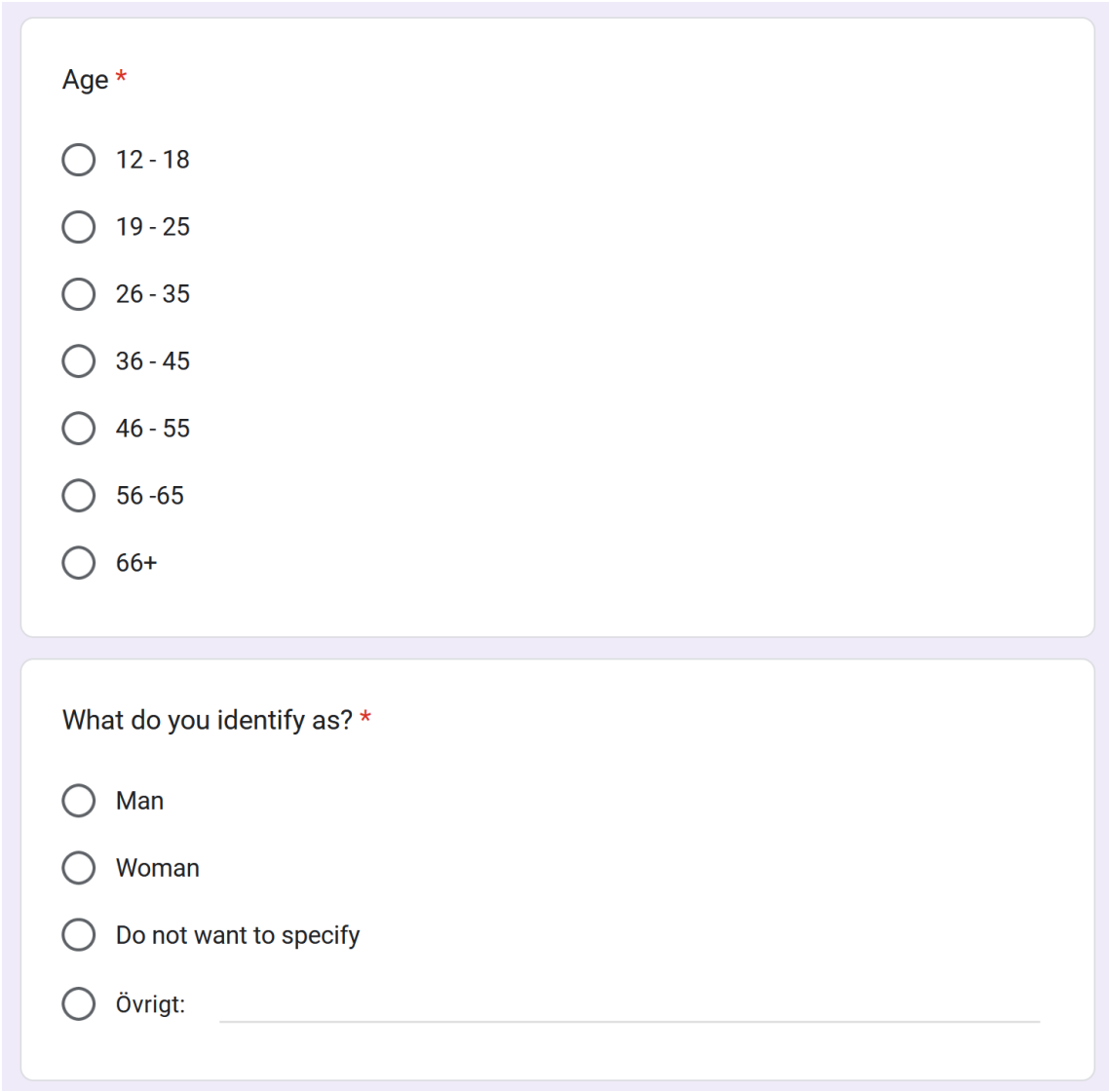

Figure B.2: Screening questionnaire section 2

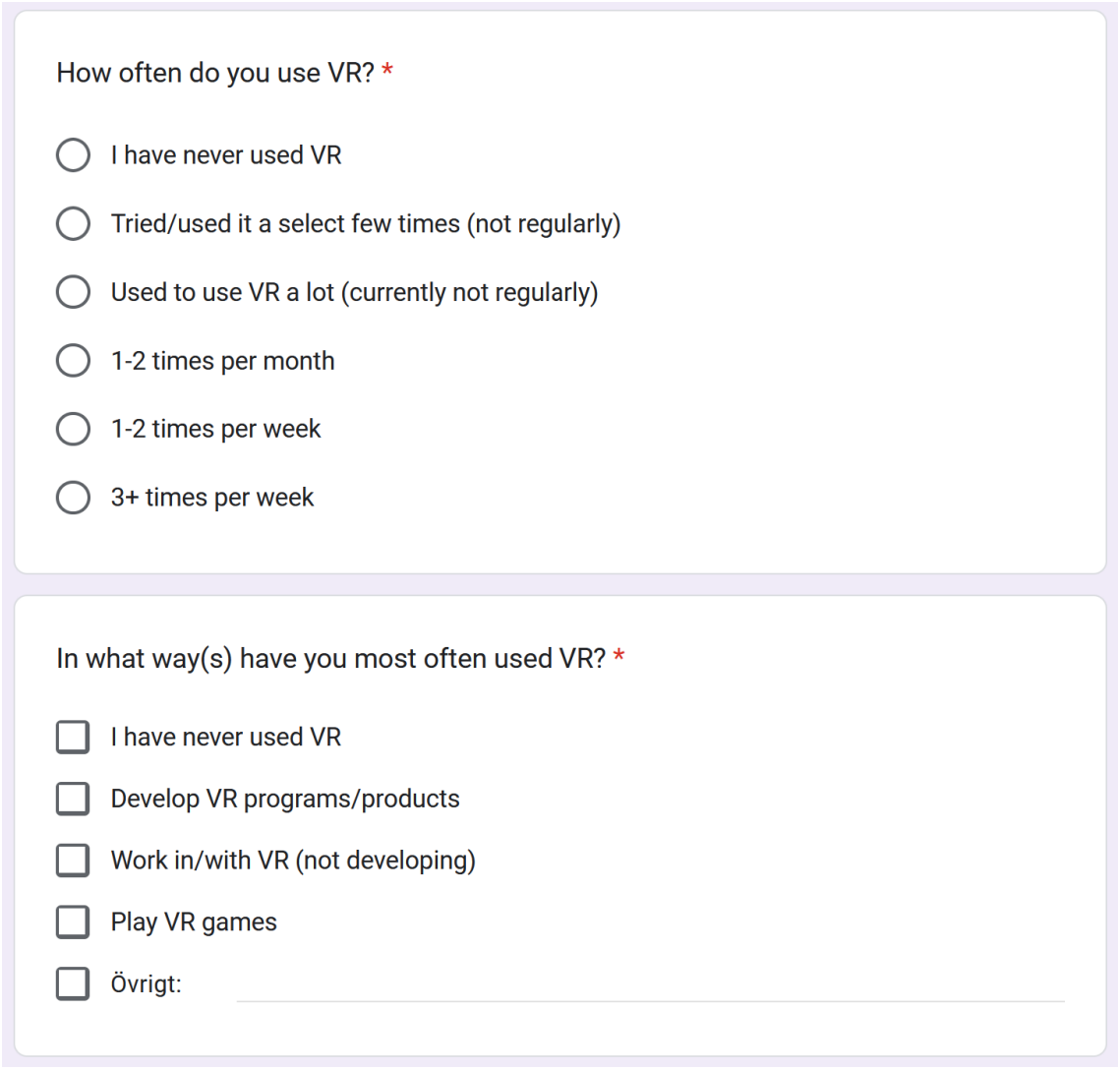

Figure B.3: Screening questionnaire section 3

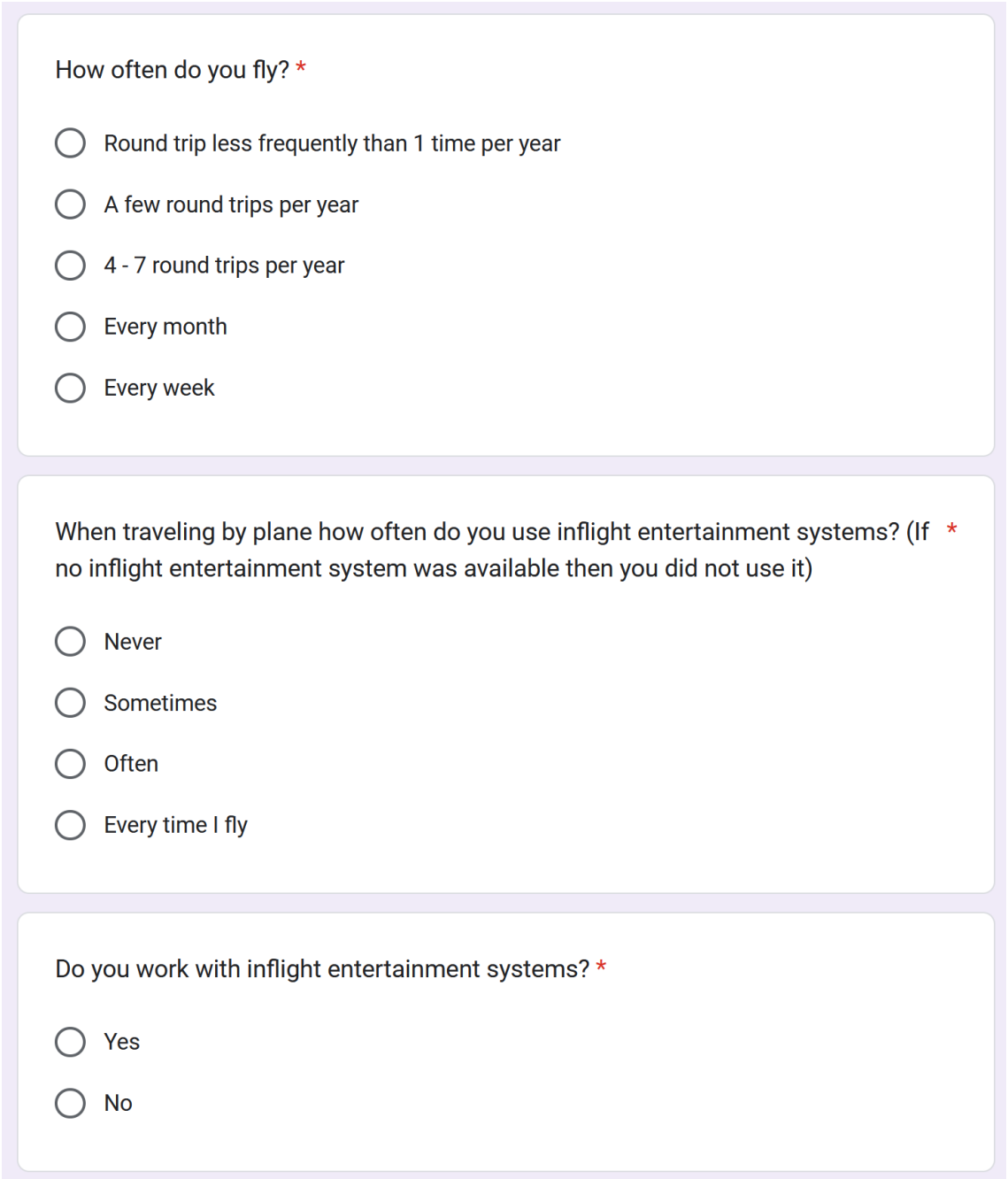

Figure B.4: Screening questionnaire section 4

B.3 Orientation script / Task scenarios

Remember:
Write their number in the questionnaires
If asked about it during test: If missing button leads to thinking they did wrong -> hand tracking is not 100 percent accurate

Welcoming brief

- Thank you for your participation, it's very valuable to us!
- It's a prototype so content and functions can be missing.
- YOU can't do anything wrong!
- Remember to think out loud, tell us what you think
- For some it helps to imagine that you have a colleague or a friend sitting next to you whom you are going to teach how to use this app.
- If you get stuck, get confused, tell us that too.
- It's the SYSTEM we are testing, not you!

Scenario
You are on an airplane that has just started its journey to "Insert destination" . You have gotten yourself settled and taken in the sight from outside of the window. But it seems to be a very cloudy day today so there isn't much to look at at the moment. You instead turn your focus towards the screen on the back of the chair in front of you and decide to see what you can do with it.

Scenario 2
Once again, you're on an airplane headed for "destination" and you want to see what you can do with the screen in front of you.

Task 1
You see that there's a lot of information on the screen about the flight but your imperial unit knowledge is a bit iffy so you would like to change it. How would you go on to do that?
if success Now you should be able to understand the numbers so you decide to check how long you have left to travel to get to your destination.

Task 2
While going through the menu you noticed a tab called "aircraft view" which you thought sounded interesting. Can you find and open that tab?

Task 3
You see that there are the names of multiple cities on the map. You're getting curious and want to see if you can find any more information about a city you like. Do you think that's possible and how do you try?

Post test questions

Have you worked with ARC maps specifically?

How did you experience the navigation between the different views and exploration of the map? (VR vs Normal not general)

How did being in VR affect your experience doing this test? Did it make it more or less interesting or didn't really matter.

Did your opinion about the product (ARC) change when being in VR vs not?

How much did you feel like you were in an actual airplane when doing the tests in VR?

How responsive was the screen?

How responsive was the hand tracking?

How accurate was pointing at the screen?

Observer Questions

Presence Questionnaire

Questions for us?

Look for questions to ask during the test. More important to follow up on specific cases from the tests

B.3.1 Iteration 2 changes

For iteration 2 we kept the same brief, tasks and scenario-descriptions. The only change was as described in the report the removal of scenario 2, the addition of VR preferences and the updated post test questions shown here

VR preferences:

Which group of passengers do you prefer? (toggle different passenger setups)

Can you tell me where the sound is coming from?

Post test questions:

Have you worked with ARC maps specifically? (question for people from the office)

How did you experience the navigation between the different views and exploration of the map? (VR vs Normal not general)

How did being in VR affect your experience doing this test? Did it make it more or less interesting or didn't really matter. Did your opinion about the product (ARC) change when being in VR vs not?

How much did you feel like you were in an actual airplane when doing the tests in VR?

Was there anything in the simulation that you would like to change? / that was weird? / that did not fit in? / that disturbed the experience?

When you were told the scenario description without VR, did you imagine your surroundings?

How responsive was the screen?

How responsive was the hand tracking?

How accurate was pointing at the screen?

Was there anything in VR that was unpleasant or uncomfortable?

Did you notice the pink dot?

If they work with usability testing: Do you see this as an alternative for usability testing?

Observer Questions

Presence Questionnaire Questions for us?

Look for questions to ask during the test. More important to follow up on specific cases from the tests

Extra question crutches

What do you think was the cause of the problems?
What did you think about … when this happened?
When this didn't work, how did it feel?
If mentioned it felt like being on the plane, was there anything that contributed the most / least?

B.4 Presence Questionnaire

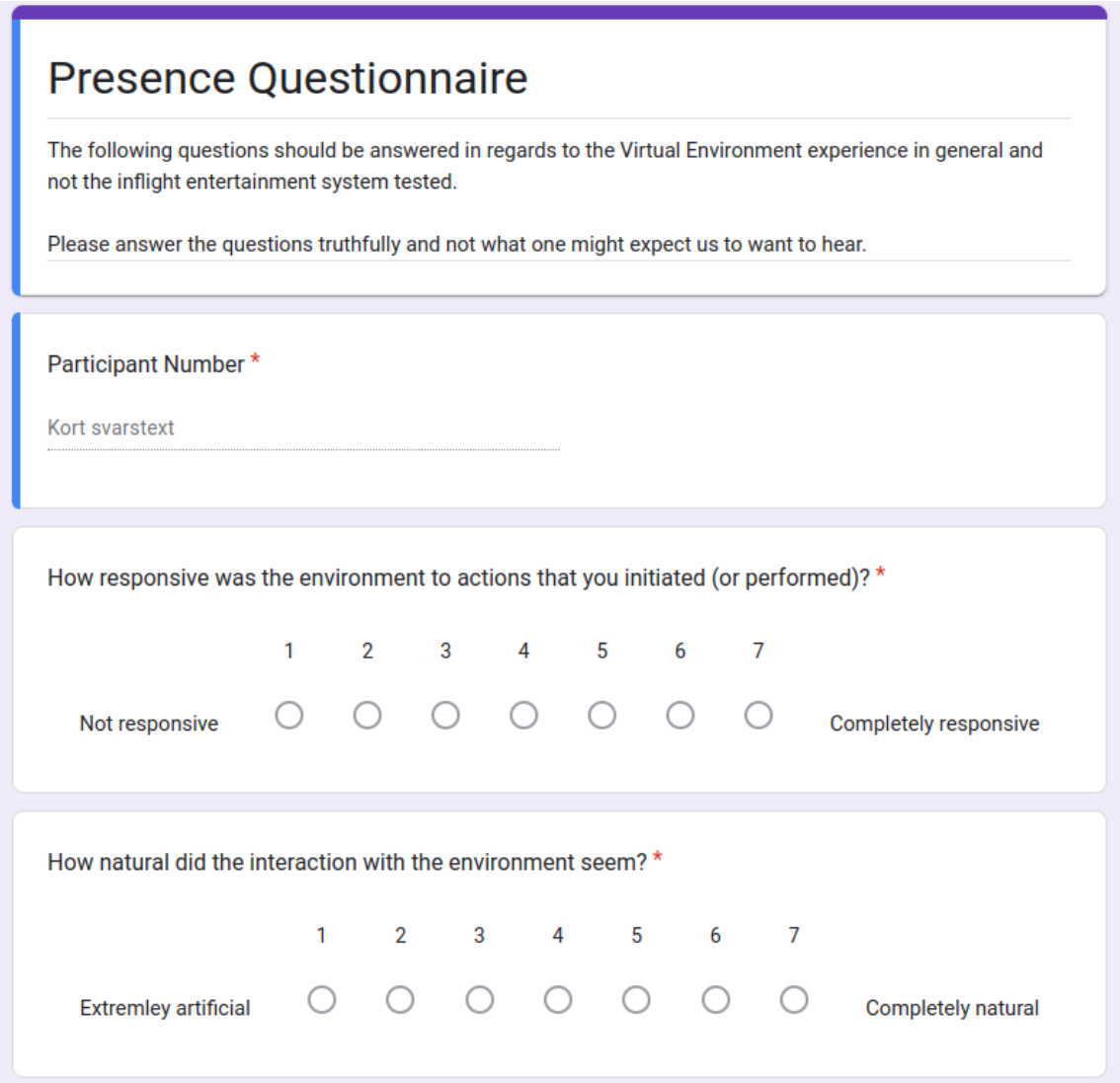

Figure B.5: Presence questionnaire section 1. Questions PQ0 - PQ2.

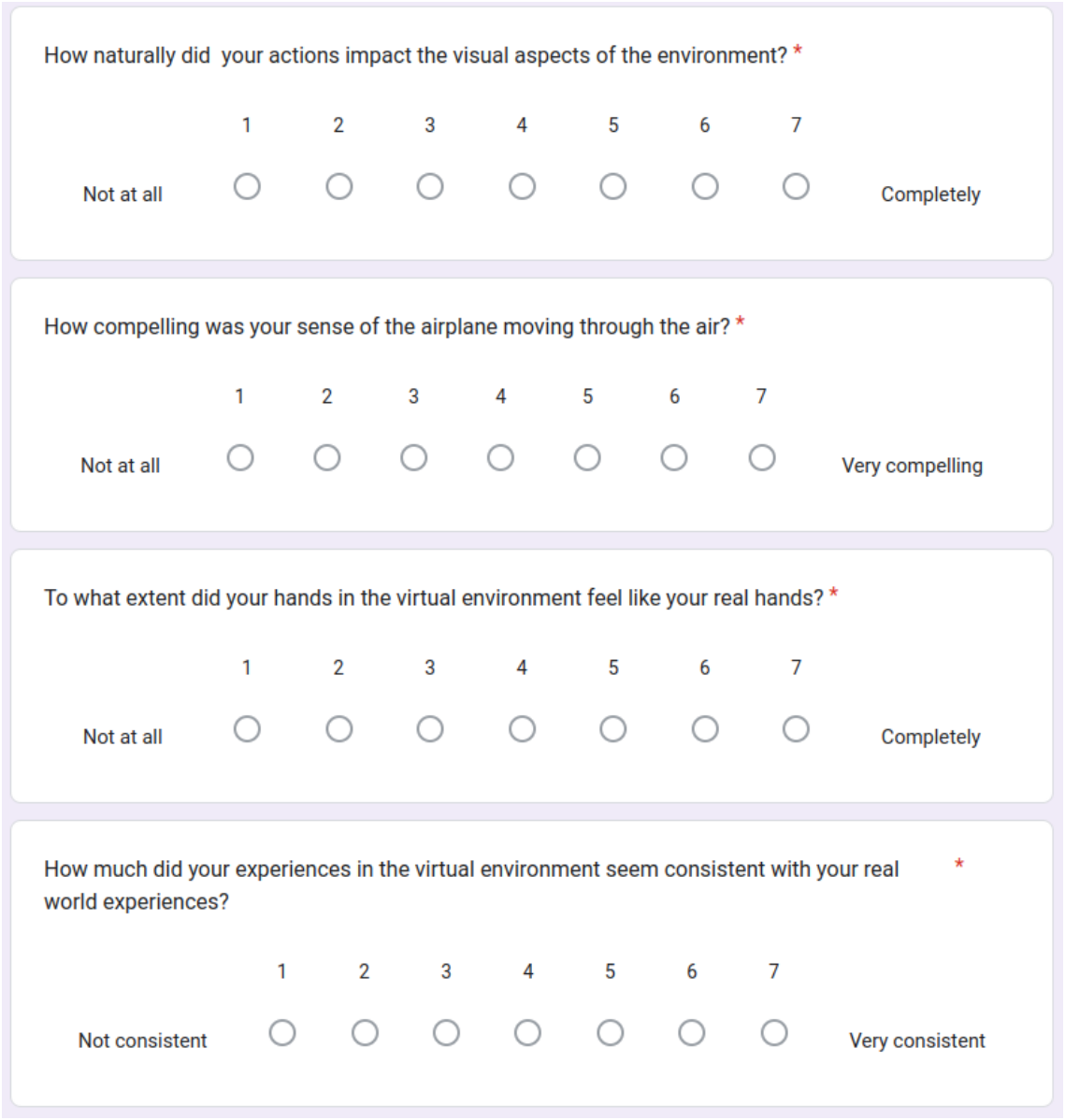

Figure B.6: Presence questionnaire section 2. Questions PQ3 - PQ6.

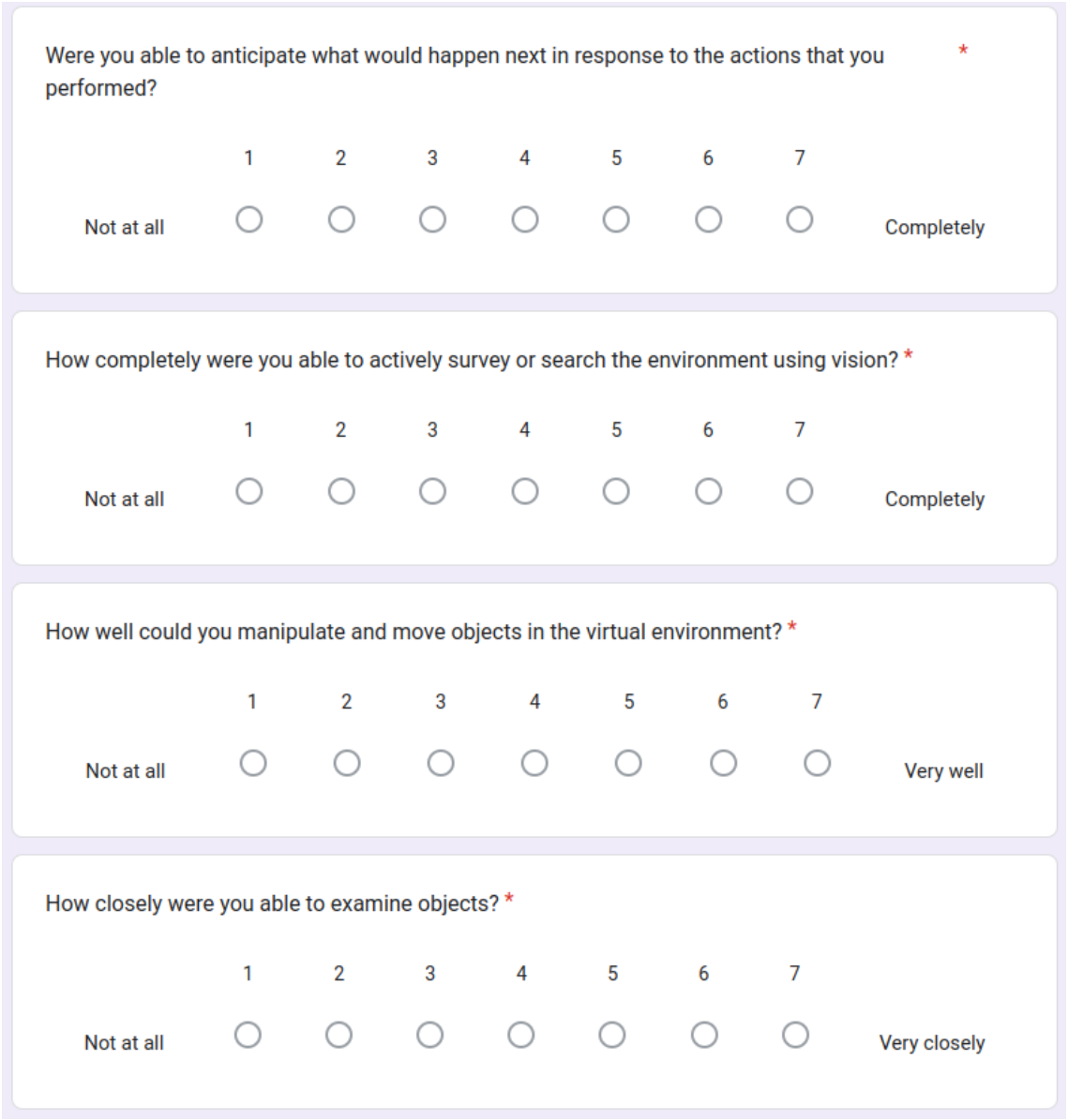

Figure B.7: Presence questionnaire section 3. Questions PQ7 - PQ10.

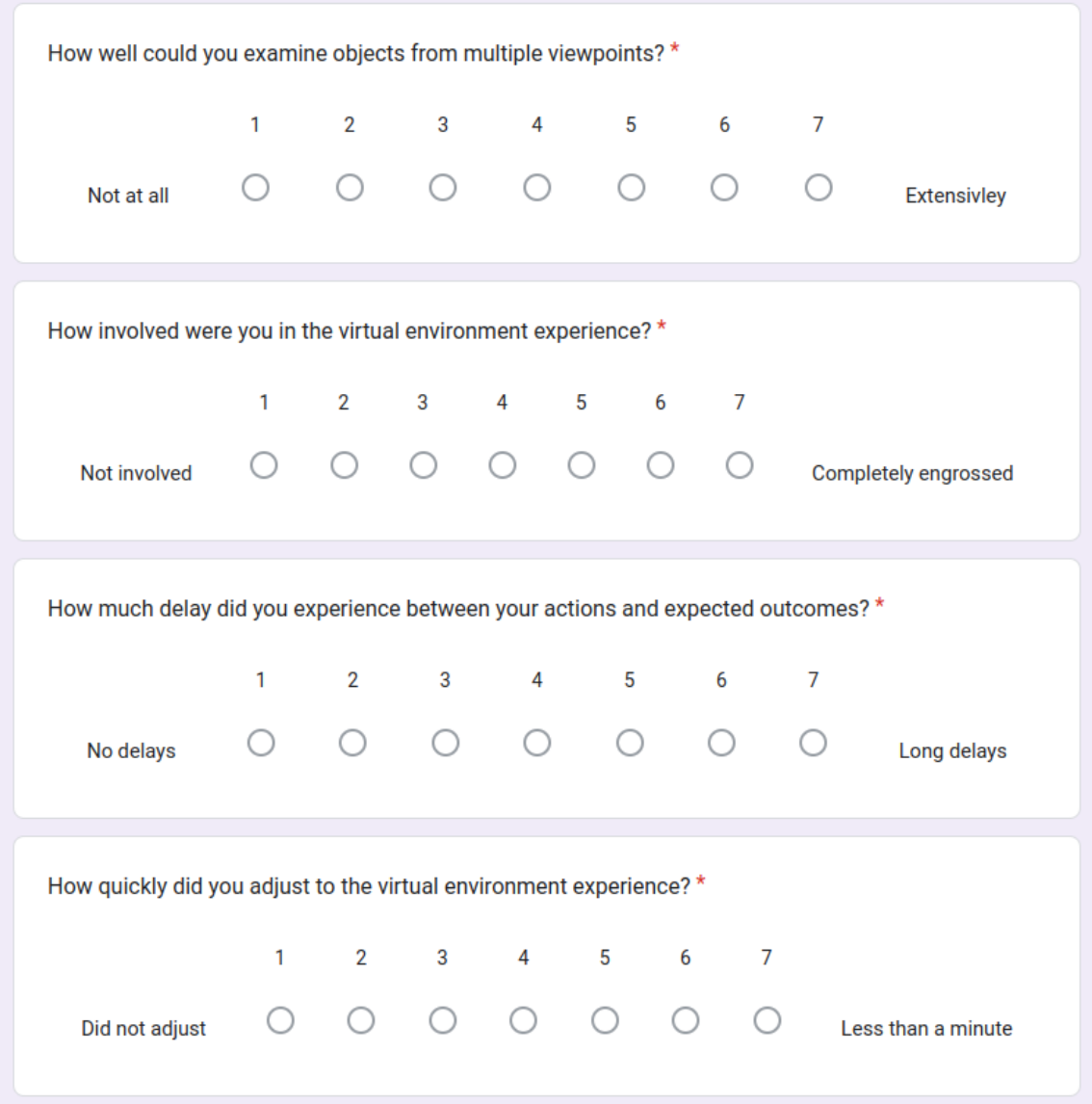

Figure B.8: Presence questionnaire section 4. Questions PQ11 - PQ14.

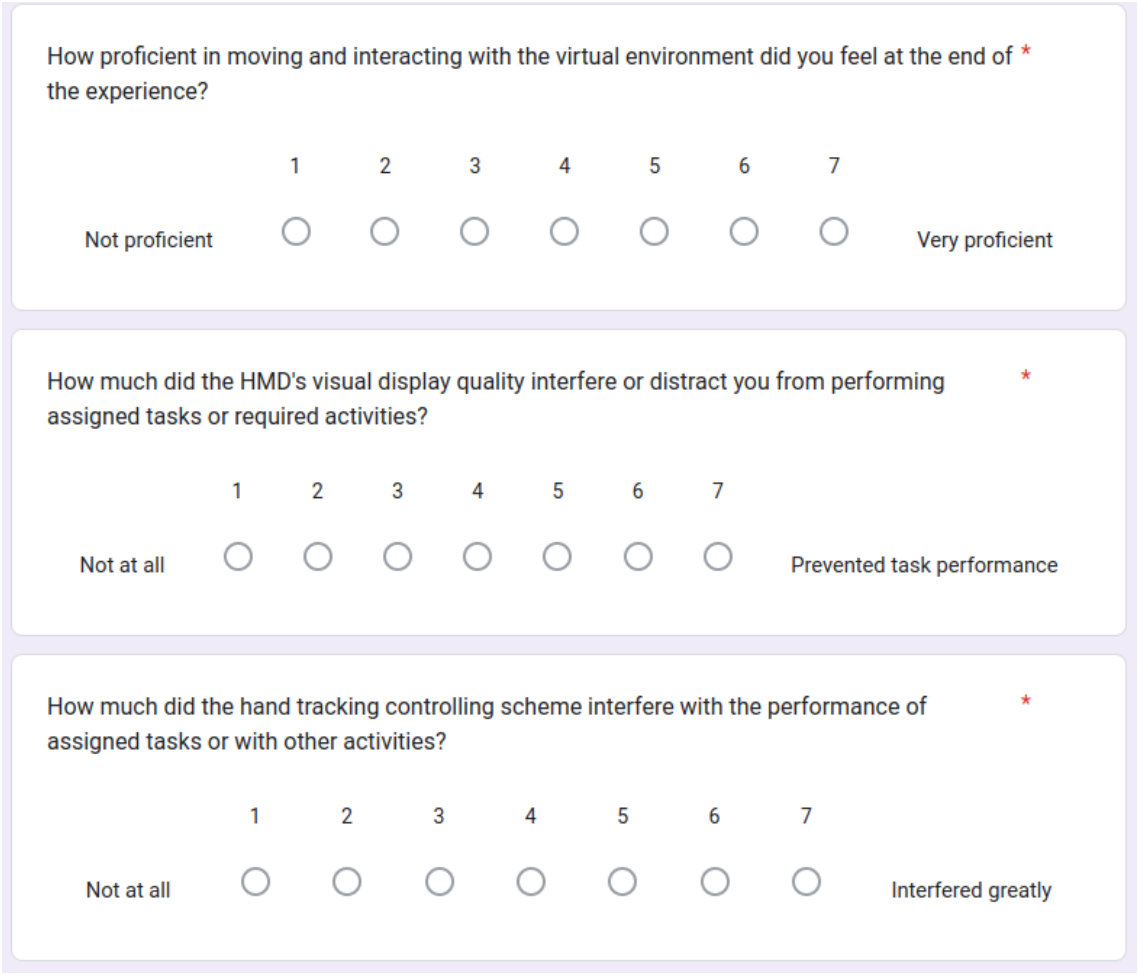

Figure B.9: Presence questionnaire section 5. Questions PQ15 - PQ17.

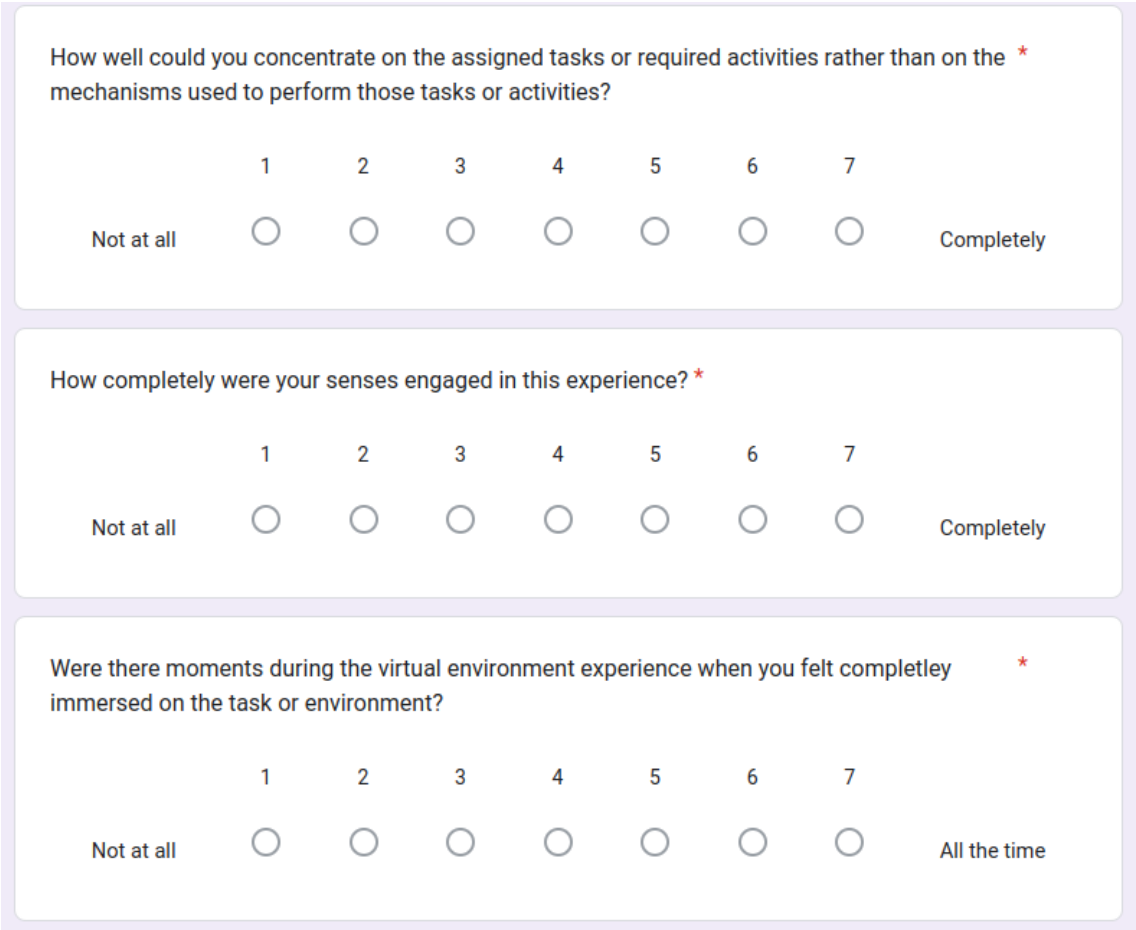

Figure B.10: Presence questionnaire section 6. Questions PQ18 - PQ20.

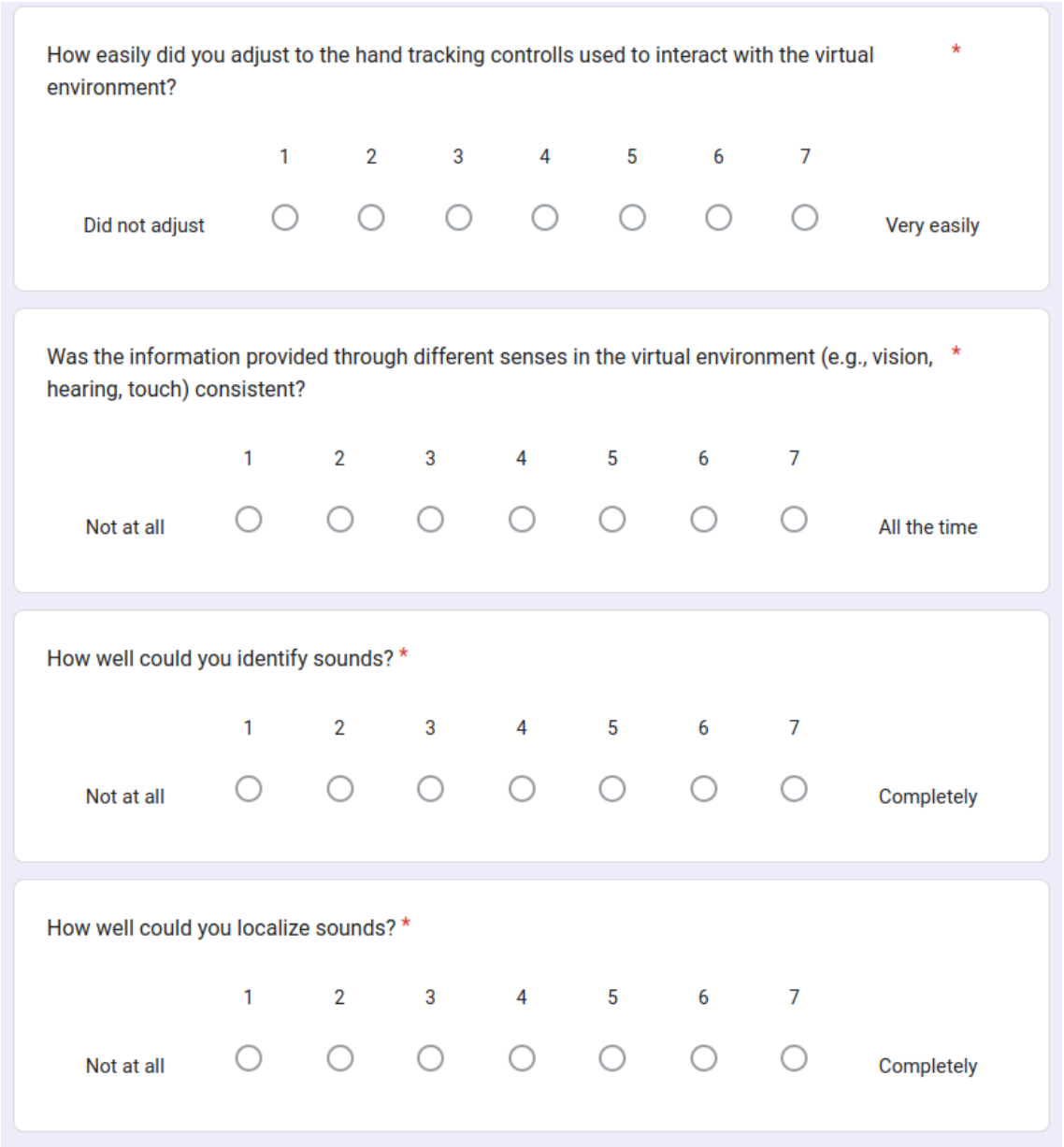

Figure B.11: Presence questionnaire section 7. Questions PQ21 - PQ24.