CHALMERS UNIVERSITY OF TECHNOLOGY | UNIVERSITY OF GOTHENBURG

Presence when interacting with Graphical User Interfaces in Virtual Reality

A Qualitative Study of Tutorials in Virtual Reality Games

Master's thesis in Computer Science and Engineering

ROBIN LILIUS-LUNDMARK
DAVID TORBJÖRNSSON

Department of Computer Science and Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
UNIVERSITY OF GOTHENBURG
Gothenburg, Sweden 2020

Master's thesis 2020

Presence when interacting with Graphical User Interfaces in Virtual Reality

A Qualitative Study of Tutorials in Virtual Reality Games

ROBIN LILIUS-LUNDMARK
DAVID TORBJÖRNSSON

UNIVERSITY OF GOTHENBURG

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Computer Science and Engineering
Chalmers University of Technology
University of Gothenburg
Gothenburg, Sweden 2020

Presence when interacting with Graphical User Interfaces in Virtual Reality
A Qualitative Study of Tutorials in Virtual Reality Games
ROBIN LILIUS-LUNDMARK
DAVID TORBJÖRNSSON

Supervisor: Thommy Eriksson, Department of Computer Science and Engineering
Advisor: Kristoffer Benjaminsson, Fast Travel Games
Examiner: Staffan Björk, Department of Computer Science and Engineering

Master's Thesis 2020
Department of Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: The scene and play area for Puzzle 1 and Tutorial C, viewed from above (see Section 6.3 for more details).

Typeset in LaTeX
Gothenburg, Sweden 2020

Presence when interacting with Graphical User Interfaces in Virtual Reality
A Qualitative Study of Tutorials in Virtual Reality Games
ROBIN LILIUS-LUNDMARK
DAVID TORBJÖRNSSON
Department of Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg

Abstract

When developing games for virtual reality (VR), presence – a sense of "being there" in the virtual environment – is an important aspect to take into account, especially when designing tutorials to those games. The reason for this is the often negative impact graphical user interfaces can have on a user's sense of presence. This master's thesis has therefore investigated what factors that needs to be considered, and how these factors should be used, in order to not break presence when designing tutorials for VR games. We have achieved this by designing and developing a VR game that contains three different tutorial interfaces, designed to induce different amounts of presence in a user. These three tutorial prototypes has been tested by VR users, and then compared and evaluated through qualitative interviews and the *Igroup Presence Questionnaire*. Finally, the resulting data was coded, analyzed and summarized into 20 general guidelines for designing tutorials for presence in VR. The result partially followed notions from previous studies, but also revealed some interesting angles on the subject. This study is entirely qualitative and has not been controlled for statistical significance, neither have the guidelines been practically tested yet, so one should be aware of the possibility of different outcomes due to differences in circumstances when applying them. Although not an entirely complete and absolute solution to the problem, we hope that these guidelines can help developers in this area, and perhaps act as stepping stones for future studies on the subject.

The prototypes can be downloaded and played at itch.io[1] and the project files are also available on Github[2].

Keywords: Interaction design, thesis, virtual reality, VR, interface, tutorial, presence, immersion, diegesis, qualitative.

Acknowledgements

We want to thank our supervisor Thommy Eriksson for his help and technical support during this project, not the least when the Covid-19 pandemic hit us all hard.

This project has also been executed in collaboration with Fast Travel Games in general and Kristoffer Benjaminsson specifically, that helped us out with both their expertise in the field of VR, and the equipment that we needed to carry out this project.

Lastly, we want to thank all of the testers that helped us and gave us feedback on the prototypes, without whom we would not have been able to finish this project. A last mention also goes out to those that wanted to test before the pandemic hit, yet didn't have opportunity to, afterwards.

Robin Lilius-Lundmark & David Torbjörnsson, Gothenburg, June 2020

Contents

List of Figures

List of Tables

1

Introduction

Virtual Reality (VR) is an immersive technology medium where the user experiences a virtual reality through head-mounted stereoscopic displays (HMDs), often also with audio capabilities and some kind of controllers to interact with the environment. It can display everything from passive and calm experiences, to fast-paced or very interactive games. Research on VR is relevant now more than ever, since new technology has created a renaissance for VR the last couple of years, and a commercial market that is currently gaining ground as this is being written[13].

In VR-technology, the user is placed inside a virtual world, and is encouraged to interact with it through head movements and handheld controllers. This creates an embodied experience with a strong sense of presence (which means that it feels like you are really there in the virtual environment, and not just observing it[14]), where the means of interaction is one of the most important and integral pieces of the technology. To design for this interaction is therefore immensely important – not only because the interactive technology in VR is in more focus than in many other mediums, but also because missteps in this area can lead to bad user experiences, which can be much worse than in other mediums.

A big factor when measuring presence in VR, is how graphical user interfaces (GUIs) are depicted, since a greater awareness of an interface can break presence[15]. This challenge has been a great incentive for many creative solutions, especially in VR games, for how to display or convey information, with examples such as *Boneworks*[9] or *Until You Fall*[8]. However, there are little to no guidelines for how to create great GUIs in VR that do not break presence, and thus, work is needed in this area.

This study will focus mainly on games in VR, partly because games tend to have intricate and complicated rule systems, which in turn need GUIs that is complicated enough to deal with all that information. There are also a lot of games to study, which makes a pre-study easier to perform. Additionally, this study is executed in collaboration with the VR game company *Fast Travel Games*, so focusing on games would give us the most benefits from this collaboration, as well.

Figure 1.1: Left: In-game button-prompt in *Celeste*[3], Right: in-game instructions all at once in *The Binding of Isaac*[4].

1.1 Research problem

When playing a game, either in VR or in other ways, some kind of tutorial will often be part of the start of that game, to show a beginner how the different game mechanics works and are meant to be used. With “tutorial”, we refer to some kind of visual instructions or GUI, often in combination with button-prompts or text-based directions to the user, either shown all at once or in sequence (see Figure 1.1).

Tutorials are often information heavy, though, and can be perceived as tiring or off-putting. In VR, this is especially a big problem, as this also can break immersion and presence and thus diminish the enjoyment of the game for a player.

1.2 Research question

Which factors needs to be considered, and how should these factors be used, to not break presence when creating tutorials for virtual reality games?

1.3 Aim

The goal of this study is to summarize what kind of factors that affects the sense of presence in VR games, by first implementing a base prototype that shows problematic ways of designing a tutorial in VR (when measuring sense of presence) and then implementing two different prototypes that improve on the base case in different ways. These prototypes will then be playtested with users to make a qualitative analysis of what works well or not, and what to think about when designing tutorials in VR games.

1.4 Delimitations

This study will not test more than two distinct ways of improving on the base prototype. Furthermore, it will not make a quantitative analysis of which of the three prototypes that works best, in regards to sense of presence, nor will it test for specific effects of different factors or interactions between them. As a result, this study will not produce statistically based conclusions – only qualitative ones.

During the playtests, only an Oculus Rift S headset[16] will be used, and no other VR headset will be tested for comparison.

2

Background

This chapter will cover an introduction to the stakeholders for this project, as well as a background to games and tutorials in VR.

2.1 Stakeholders

These are the stakeholders for this Master's thesis and project.

The Authors

We, the authors of this report, Robin Lilius-Lundmark and David Torbjörnsson, are considered as stakeholders for the development of the prototype, the performance of the research project and the writing of this thesis; as we are singularly responsible for all of these parts.

We are both doing this project under the Master's programme of Interaction Design and Technologies, at Chalmers University of Technology in Sweden. This Master's programme has a large emphasis on the design for interactive experiences and how to best design a user experience. Courses taken in this Master's programme, like for example: Technology-Driven Experimental Game Design, Designing User Experiences, Graphical Interfaces and Gameplay Design; should be very helpful to have in mind during this project.

Fast Travel Games

This project is performed in collaboration with the VR-focused game studio *Fast Travel Games* in Stockholm, that have agreed to assist with technical equipment, as well as technical and theoretical guidance, and software examples to work with. *Fast Travel Games* first debuted with the game Apex Construct in 2018, and later released *The Curious Tale of the Stolen Pets* in 2019. At the same time, the game *Budget Cuts 2* was released, which was developed by *Neat Corporation* in partnership with *Fast Travel Games*, as well.

In this collaboration, Kristoffer Benjaminsson at *Fast Travel Games* will act as contact person and mentor. Kristoffer is one of the original founders of the company and currently has the position of CTO (Chief Technical Officer).

Fast Travel Games first and foremost are interested in the results of this project, in terms of useful insights or guidelines when it comes to working with graphical user interfaces in VR, without breaking presence.

Thommy Eriksson

The academic supervisor of this project is Thommy Eriksson, who work in the Interaction Design Division at the Department of Computer Science and Engineering at Chalmers University of Technology. Thommy has a PhD in Digital Representation and his research focus mainly on ICT and learning, and Mixed Reality. Right now he also has the role of Director of Master's programme Interaction Design and Technologies.

2.2 VR games

There is a lot of different genres of games in VR, everything from puzzle games to first person shooters, and a lot of ways to control these games. A review of different relevant games (some can be seen in Figure 2.1) was performed in the pre-study of this project, and can be viewed in Section 6.1.2.

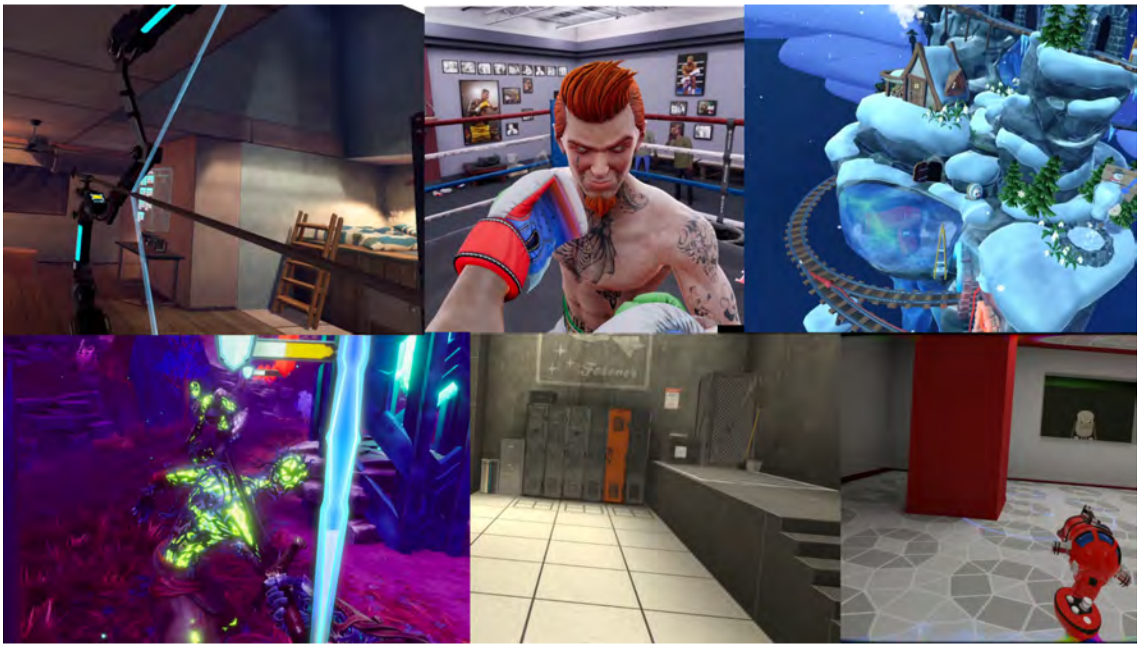


Figure 2.1: Games from top left: *Apex Construct*[5], *Creed: Rise to Glory*[6], *The Curious Tale of the Stolen Pets*[7], *Until You Fall*[8], *Boneworks*[9], *Budget Cuts 2*[10].

A big limitation with VR games is that the extremely immersive nature of VR actually can lead to users feeling so-called cybersickness which is the opposite of normal motion sickness, but with the same nauseous results. This is often caused by moving around in VR while standing still in the real world, since this creates a big contrast between how the body's inner ear register movement and what the eyes see, thus causing motion sickness. Other causes include bad framerate or if the HMD is not optimally configured[17, 18]. Many different attempts to get rid of cybersickness, focusing on virtual movement, has been tried, with varying results. The following movement models are the most common in VR today:

In some games, you can move around freely with continuous movement like a non-VR game, as in *Boneworks*[9]. In others, you can teleport around by pointing a controller to choose a new spot, as in *Budget Cuts 2*[10]. There are also games where you are completely or mostly stationary, like *Beat Saber*[19]. Some games even combine different ways of moving around, such as *Until You Fall*[8], that use slow continuous movement combined with teleportation, where the latter doubles as a dash-attack as well.

Since mainstream VR games are relatively new, not a lot of standard implementations of the controls have been decided upon by the industry and it's up to the developer to find out what works for a particular game. However, many games let the player have hands that maps to the controllers, and then encourage the player to pick different objects and tools up, such as *The Curious Tale of the Stolen Pets*[7] or *Boneworks*. Other games replace these hands by either the actual controllers you are holding, as in *Google Earth VR*[20], or tools instead of hands, such as in *Beat Saber*.

Figure 2.2: Tutorial screen on a TV in *Boneworks*[9].

2.2.1 Tutorials

Tutorials in games are necessary to teach new players the controls and mechanics of the game. The way a tutorial is implemented could differ a lot between games and should not only teach the things that are special for this game but also teach the very basic controls to help players that don't yet have learned these. This is especially important when interacting with new ways of controlling a game such as novel technology as VR often are to new players, with the HMD and controllers.

In ordinary non-VR games that players view on flat screens, rather than in immersive HMDs, tutorials has been used for a very long time, so there is a long tradition and many different standard ways of doing it. Most games use non-diegetic (see Section 3.4) GUIs in sequence to present information to players when needed, but there are of course also a lot of variants and different versions of this. In *Celeste*[3], players get in-game button-prompts when needing to learn new functionality, while in *The Binding of Isaac*[4], all the controls are given at once. Some games, such as *Tomb Raider*[21], lack instructions but instead provide a safe environment to experiment and learn new abilities, as a different kind of tutorial, before putting the player to the test. Something that is rarely used anymore, is having instructions outside of the game in a pamphlet or on the side of the arcade cabinet, as *Pac-man*[22] did.

Because of the very different ways of both experiencing and controlling virtual environments in VR, there are no real tutorial standards for VR yet. It's natural that many VR games has tried to use similar tutorial approaches as for ordinary games, but are often met with mixed results, since flat, non-diegetic signs can be off-putting if not done right. Moreover, in contrast with normal games, a player cannot see their own hands or the controllers they hold, making it harder to check if you're pressing the correct button.

Figure 2.3: Tutorial screen for blocking in *Until You Fall*[8].

One successful way to do it seems to be to put small notes or signs pointing at a representation of the controllers that you are holding, in the virtual environment, to explain different buttons and functions. This is utilized by *Google Earth VR*, for example. A similar but different method is to put non-diegetic signs in the otherwise diegetic environment itself, to point out important things that the player might need, as they do in *Apex Construct*[5]. In *Boneworks*, the tutorial is actually a museum with lots of showcases and diegetic interfaces (see Figure 2.2) that show you both instructions and interactions, as well as put you through different challenges, to teach you. In *Until You Fall*, on the other hand, successfully use the aforementioned non-diegetic, flat screens with more text (see Figure 2.3), instead. Like in non-VR games, the kind of tutorial, how it looks and what they teach is very different between games.

3

Theory

This chapter contains relevant theory on the subjects of immersion, sense of presence, narrative involvement and the design of tutorials, especially in VR.

3.1 Presence

Witmer and Singer says that “Presence is defined as the subjective experience being in a place or environment, even when one is physically situated in another”[15].

Presence is the subjective human response to an immersive environment (the sense of actually being there)[14, 23]. When interacting with the environment and things responds to that interaction in a way that the user thinks they should behave, it enforces the feeling that the user is actually there.

There are also several factors that could increase the sense of presence, like spatial audio[24], sense of depth[15] and also emotional investment[25]. See Section 3.1.2 for a summary of all factors that will be used in this study.

3.1.1 Measuring presence

There are three major approaches for measuring presence generally discussed and used in relevant literature: subjective, physiological and behavioral. Mostly, subjective measuring means some kind of post-study questionnaires. Physiological measures are more objective in kind, and make use of special equipment attached to the subjects to measure real-time data. Behavioral measures, on the other hand, focus on observations of how a subject behaves as a consequence of different stimuli in the virtual environment[26].

Three big questionnaires that are mostly used in this research field[26, 27] are the *Presence Questionnaire* by Witmer and Singer[15], the *Igroup Presence Questionnaire*[28] and a questionnaire by Slater, Usoh and Steed[29]. See Section 4.3 for more details on which questionnaire that was chosen for this study.

Schwind et al. suggests that to avoid breaks in presence[30] when measuring presence through the use of questionnaires, one can digitally implement the questionnaire in the virtual environment. This would mean that the participant should not need to take off their HMD and thus have a higher chance of keeping the sense of presence that is supposed to be measured[27].

3.1.2 Factors that influence presence

To be able to make informed decisions on what to focus on in this project, an *Affinity Diagram*[31, p.12] analysis based on different literature was performed during the pre-study (see Section 4.1). This analysis revealed a collection of different factors that seemed to influence presence, although these different categories are not exclusive and overlap at times.

Body realism

A good virtual body representation[24] or multimodal presentation[15] increase presence, while unrealistic bodies or body movement decrease it[32].

Control

More fine-tuned control, or just more control, over your virtual self is generally better for presence,[33, 15, 24, 23], while difficulties when moving around may decrease it[32].

Diegetic sound

Spatial sound increase presence[33, 24, 23], while non-diegetic music decrease it[34].

Expectations

When the virtual world behaves as you'd expect from your experiences with the real world, it increases presence[15, 32], and the opposite instead decreases it[32].

Immersion

There are several physical or technical aspects that facilitate the possibility of immersion, and increasing immersion generally leads to better presence[33, 24, 15, 30, 32, 23].

Interface awareness

Being more aware of virtual interfaces than not, may decrease presence[15].

Involvement

When the content of a virtual environment, like its characters and narrative, are interesting and draws you in, it increase presence[25, 15, 32, 23].

Negative emotions

Negative emotions may both increase or decrease presence: being embarrassed or self-conscious when in VR may decrease presence, while anxiety about something in the virtual world might increase it[32]. Negative emotions in VR can create a reaction that decreases presence, as well[25, 30, 32, 35]. In summary, this is without a doubt a factor that affects presence, but it's not as simple as claiming "negative emotions decrease presence”.

Physical realism

When the virtual environment behaves realistically and objects are physically simulated[33] and interactable it increases the sense of presence[15]. Restricting these interactions can instead decrease the sense of presence[32].

Shutting out the outside

When you’re able to shut out the real world outside of the virtual environment, for any reason, presence may increase[33, 32, 15], while distractions that focus your attention on the real world and away from the virtual environment, can decrease it[33, 30].

Visual realism

Realism in virtual environments, but not necessarily in the shape of photo-realism, increases presence[33, 24, 34, 15, 32], while flat[34] or non-rich environments decrease it[15].

3.2 Immersion

Immersion is the physical attributes or technical solutions that allows a user to immerse themselves in a virtual environment. This includes head-tracking responsiveness, framerate, and spatial sound, among other aspects according to Slater[14].

As technology becomes better, with constant or higher framerates, and tracking gets more accurate and responsive for both HMDs and controllers, it gets easier for users to be immersed. These aspects can be objectively measured and are easily verifiable to be better[14, 23].

Witmer and Singer disagrees with this objective look on immersion and describes it as follows: “Immersion is a psychological state characterized by perceiving oneself to be enveloped by, included in, and interacting with an environment that provides a continuous stream of stimuli and experiences”[15].

For this study we decided to go with the definition by Slater to give a clearer distinction between presence and immersion.

Due to the nature of VR, immersion is also the cause of VR motion sickness, or cybersickness, which can cause users to experience much the same nauseating symptoms as normal motion sickness. Some people are very susceptible to this, while others are not. The cause of cybersickness is mainly sensory cue conflicts, but HMD configuration and framerate, among other things, could also be contributing factors[17, 18, 23].

Developing immersive technology was very important before the modern HMDs such as *Oculus Rift* and the *HTC Vive* when the framerate, head-tracking and resolution wasn't as good as it is now[34, 27]. Recently, though, with new and more immersive technology, focus has shifted to what we can do inside the virtual experience to captivate users even better.

According to Slater's definition of immersion, presence is in this context the subjective human reaction to perceiving immersion[14]. Mostly, problems with immersive aspects impact presence negatively, such as bad framerate or the design of an HMD letting in outside light which might distract the user. Immersion is the sum of all physical aspects that facilitate the possibility of immersion, and in turn, possibly presence[33, 24, 15, 30, 32].

3.3 Narrative and involvement

According to Witmer and Singer, "Involvement is a psychological state experienced as a consequence of focusing one's energy and attention on a coherent set of stimuli or meaningfully related activities and events" and when the user gets more involved in the virtual environment, this leads to a greater sense of presence[15].

Narrative and involvement are also separated from presence in that you can be very emotionally involved in a story but not feel that you are there. You can also feel like you are in a virtual environment but not be interested in what happens there[14].

The above is further supported by a recent study, which showed that a key to narrative engagement and empathy in a narrative is the user's belief about their role in a story. If the user believes they are just observing something – even when feeling presence and being there in the virtual environment – they feel less empathy or engagement, compared to believing that they're part of the narrative[36].

Studies of the emotions of players in games shows that positive emotions correlated with higher presence[37, 25] while negative emotions can, but do not necessarily, do the opposite[25, 30, 32].

There might be some connection between involvement and the feeling of presence, since users in a study by Riches et al. reported higher levels of presence when being involved emotionally, even in some cases when the emotional response was negative, such as feeling embarrassed or lonely [32].

Another study examined the effect on presence of being more emotionally involved in the story of a game, by giving one of two test groups a backstory before playing, and comparing their presence scores afterwards. The result supports Riches et al. above, in that there is a trend that more story involvement boosts presence. However, in contrast, these results were not statistically significant[38]. Brooks agrees by arguing for the importance and potential of engaging narratives for presence and immersion, but did not statistically prove anything either[39].

The concept flow is tangential to involvement, and somewhat also to presence. It's the the state of "being in the zone", of forgetting everything else when focusing on a certain task, and is often triggered when the balance between challenge, ability and reward is well balanced and perceived as “just right”[40]. However, although similar concepts, it seems like presence is a cause of flow, rather then the other way around[41].

3.4 Diegesis

Diegetic UI or game elements are things that are actually placed in the virtual world that virtual characters can see and interact with, while non-diegetic elements are things that the player can see but are not part of the virtual world and the characters themselves can not see (like a health bar or ammo count)[42] (see Figure 3.1).

Figure 3.1: Non-diegetic user interfaces of *Team Fortress 2*[11] and diegetic user interfaces of *Dead Space*[12].

By removing non-diegetic elements of a heads-up-display (HUD), the gameplay experience could be improved according to Iacovides et al.[43]. While this might not be directly connected with presence, the more enjoyable the experience is the easier it should be to feel presence.

A study by Salomoni et al. argues that diegetic interfaces in VR increase, not only a sense of presence, but immersion, usability and cybersickness avoidance sensations[44, 45]. Since it was a smaller study with only 10 participants it’s hard to draw any general conclusion from this, or know if it has the same effect with other interfaces such as tutorials.

Diegetic elements are not only visual, as they can also be auditory. Spatial and diegetic sounds also have been shown as increasing presence[33, 24] and non-diegetic such as background music decrease it[42, 34]. Interestingly, in a study by Nunez, apparently matching the content of non-diegetic music with the virtual environment can turn it into a positive enforcement of presence, instead of the opposite[35]. This is however not very relevant to the current study.

3.5 Tutorials

There are not a lot of research done in the field of tutorials in games and especially not in VR. Many of the same ideas from tutorials in non-VR games may still apply to a tutorial in VR, though.

Tutorials are a kind of learning scaffolding often applied in games to familiarize a player with the goals and interface of a game world, in order to make a game experience as enjoyable as possible. Shannon et al. suggest following these three guidelines, when creating effective tutorials[46]:

- Immediate, positive and cognitive feedback that combines corrective and affective support
- Short bursts of just-in-time instruction with visual cues and minimal text
- Step-by-step scaffolds that fade into free play over the course of the exercise

One can divide tutorials into two categories: the traditional context-insensitive ones, and the more dynamic context-sensitive ones (see Figure 1.1). A context-insensitive tutorial is like a rule book where all the instructions are given to the player in the beginning, while a context-sensitive tutorial gives the player the instructions in the context of a game, when they need them. Context-sensitive tutorials are often shorter and generally easier to take in, which could be more enjoyable and lead to a better experience than with a context-insensitive approach[37, 47].

Context-sensitive tutorials may not necessarily lead to a better sense of presence directly, but the effects of more positive emotions and less negative emotions that these can lead to, may still increase presence[25].

4
Methodology

This chapter will describe the different methodologies that have been applied, or is planned to be used, during the project's different stages. These stages ranges through a pre-study and planning period, a prototype development, playtests with data collection and finally analysis of the gathered data.

4.1 Pre-study

To get to know the field of VR and specifically areas around the concepts of Immersion and Presence in VR, a basic *literature review*[48] will be performed during the pre-study. An alternative to *literature reviews* are *Systematic Literature Reviews*[31, p.40] or *Systematic Mapping*[49], although these methods are much more time consuming and thus to be included in this project, a larger time frame would have been needed. It was therefore decided to stick with the basic *literature review*, in order to have time for development and testing.

There will be a lot of different factors that might increase or decrease a sense of presence in a virtual environment, proposed by different papers. To get a grip on all of these, an analysis of them will be performed through the use of an *Affinity Diagram*[31, p.12]. This will generate a larger understanding for the subject in us, while also giving a good pointer to what factors that might be best to focus on in this study.

Besides the *literature review*, a game study will be performed as well. This study will not follow any specific methodology, since the plan basically is to just make sure to have a general understanding of the field and play a lot of relevant games. However, methods such as *deductive content analysis*[31, p.40] and *casual observation*[31, p.120] might be useful to consider during this game study.

4.2 Prototype development

Three prototypes will be developed with VR-support, where participants will experience three different versions of basically the same base environment. Prototype A will act as a baseline and thus should be deliberately designed to generate a low presence score. Prototype B and C will be improvements on A, in different directions, and so these should therefore be developed after prototype A is complete enough.

To hurry up the development of the prototypes as much as possible, a well known game engine will be used, as opposed to developing the whole application from scratch. There are currently two powerful and reliable game engines available today, that support cross-platform development, a wide range of different operating systems as well as VR headsets: Unreal Engine 4[50] and Unity 3D[51], and the one chosen for this project is Unity. The reason for this is simply because the authors of this thesis are more comfortable using Unity, and that Unity-projects has had a tendency to remain much smaller in hard disk space than Unreal Engine-projects, in the past.

When developing and play-testing the different prototypes, an HMD of the model Oculus Rift S[16] will be used. This is the latest computer driven Oculus HMD as of yet. The predominant reason for choosing this HMD over other brands was availability, as Fast Travel Games had this HMD available to borrow at the time.

The main programming language in Unity is C#, and again, to make development smooth and fast, integrated development environments (IDEs) will be used. The IDEs Visual Studio Code[52] and JetBrain's Rider[53] will both be used for programming, and the reason behind using two different kinds is again comfort, since the authors of this thesis have different preferences. Luckily, using different IDEs does not impact the quality of the generated code or application in the least.

When collaborating in programming, it's common to use some kind of version control software. Among many different ones, Git[54] has been chosen, mainly because it's free and openly available, but also since the authors does not have much experience with other similar software. The most common host for Git repositories is GitHub[55], and will therefore be used to host the Unity data of this project as a backup.

The development itself will follow some of the guidelines that Agile development[56] is all about. Mainly, to structure the work, User Stories and Daily Meetings will be used, and the development will be Incremental and Iterative. This does not follow a certain Agile method, such as Scrum[57], but is rather based on the more general Agile Manifesto itself. Since it's decided that GitHub will be used, the Kanban[58] feature that the site offers can therefore act as a tool for keeping track of the User Stories.

4.3 Data collection

Due to the scope of this project, in combination with the subjective nature of the area of focus, a quantitative analysis will be hard to produce. It is still possible to achieve by having a lot of participants in the playtests and analyze them statistically, but the limited time frame of the project makes it hard to ensure that there are enough participants to make statistically significant conclusions. Thus, it is decided to focus on a qualitative approach from the beginning, instead.

With such a subjective topic, a mixed methods approach, namely *integrated stud-*

ies[59, p.277], will be applied here. This means that several different data collection methods will be utilized, and the choice of which is grounded in the research question and problem. It's important to emphasize that this does not mean that the more methods that are used, the better. Instead, it's better to choose a small number of methods that compliment each other.

4.3.1 Playtests

As the concept of presence among other things is defined as the human response to an immersive environment[14], and the research question refers to the amount of presence users experience or not: user tests (or playtests, as they are often called when regarding games) are crucial for measurement. During these playtests, each participant will be observed[59, p.271][31, p.120] and recorded, so as not to miss anything important. Specifically, critical incidents will be of interest during observation.

Before starting out with the playtests, a pilot playtest will be conducted to make sure that everything works as intended, as well as testing the mix of observations, questionnaire and interview. If nothing is amiss, this pilot playtest will be included in the rest of the study.

4.3.2 Questionnaire

Each participant in the playtests will get to test all of the three different prototypes. After each prototype has been played through, the participant will be prompted to fill in a questionnaire[31, p.140] that is intended to measure how much presence that participant experienced. Although statistical significance may not be possible, the comparable levels of presence between participants as well as between prototypes will compliment the observations well.

Among all the different questionnaires that exist to measure presence, the choice has been between the three that are used most frequently in this area of research[26, 27]: the *Presence Questionnaire* by Witmer and Singer (WS)[15], the *Igroup Presence Questionnaire* (IPQ)[28] and a questionnaire by Slater, Usoh and Steed (SUS)[29].

The WS questionnaire is considerably longer than the others (32 questions over 14 for IPQ or 6 for SUS) and did not contribute enough in other ways to make up for its considerable length. The WS questionnaire was therefore not selected. Furthermore, according to Schwind et al.[27], the IPQ questionnaire was the one out of the three that best reflects the construct of presence. When comparing the SUS and IPQ questionnaires, it was also found that IPQ covered every question in SUS – regarding meaning of content, if not literally – except for SUS question 5, and that question was considered especially hard to understand. As a result, the IPQ questionnaire was chosen to be the single one used in this study.

The *Igroup Presence Questionnaire* is divided in three subscales, except for one question that lacks subscale but instead focus on a general sense of presence. The subscales are defined as following[28]:

- **Spatial Presence:** A sense of being physically present in the virtual environment
- **Involvement:** Measuring the attention devoted to the virtual environment and the involvement experienced
- **Experienced Realism:** Measuring the subjective experience of realism in the virtual environment

When a questionnaire is being filled in, there is the possibility to let participants do this digitally while still in VR, instead of having to take off the VR-headset each time[27]. Although this would have been a more time efficient way for the participant, and possibly would lead to less breaks in presence when taking off the headset, it was decided to not use this method. This was because the development phase of the project already will be stressful, so adding another big design task would not be viable, planning-wise.

4.3.3 Interviews

To fill a final qualitative gap in the data collection, *unstructured interviews*[59, p.93][31, p.102] will be held at the end of the playtests, to follow up on interesting observations or questionnaire answers. The interviews will be casual and open, but should be kept as short as possible, due to the large amount of time that is needed to transcribe and code interview data. This is also another reason for combining interviews with observations and questionnaire, rather than relying solely on interviews.

There are alternative ways of performing interviews, than the regular unstructured way that is chosen for this study. For example, *stimulated recall*[59, p.117] is a way of supporting the participant's recall of the playtest through the use of some kind of recorded media. Then there are *focus groups*[59, p.133][31, p.92], where a group of participants, lead by a moderator, discuss a topic among themselves. There are also the *Repertory Grid Technique*, where participants are interviewed by first somehow introducing different duality constructs to the subject, and then letting the subject put the elements of the study on a scale in each construct. All of these alternatives, however, have been discarded in the interest of time.

4.4 Data analysis

The qualitative parts of the collected data, namely the observed critical incidents and the transcribed interviews, will be put through a *deductive content analysis*[31, p.40]. That means that the interviews and observations will be coded according to a specific set of predefined themes, that is going to be based on different factors that might affect the enforcement of, or the breaking of, presence (see Section 3.1.2 for a description of these). As the data is collected, the *content analysis* will progress iteratively. There is a possibility that themes or categories that weren't planned for, will emerge during this analysis. This is not a problem, but rather something to be aware of and could give interesting results.

After the *content analysis*, a custom made analysis will be performed. This one is based heavily on the *Kano Analysis*[31, p.106], but in this case, the previously mentioned themes (see Section 3.1.2) will be used as scales instead. The reason for this approach is to explore how these different factors impact the general experience of presence, when it comes to the specific GUI that is studied in this project.

The IPQ questionnaire data will serve as a way to compare the qualitative results with a regular baseline of presence scores. However, the data will not be analyzed statistically, and thus cannot be used for making statistical conclusions.

A *Grounded Theory* process[59, p.309] will not be used in this study, because that method has the aim of developing completely new theory, while this project already is based on existing theory.

5

Planning

As the planning changes during the project, this chapter will first describe the initial plans when coming into the project, before the pre-study. Since the majority of the plans and important decisions were made during the pre-study, the rest of this chapter will rather cover those plans.

5.1 Initial plans

These are the initial plans that we brought with us when starting this project.

Comparing to the later, more researched and refined plans in the following sections, it is clear that these initial plans were somewhat over-ambitious.

5.1.1 Order of process

1. Write planning report (where we describe and motivate methods, choices and plans throughout the project), while making sure to consider each decision thoroughly (with regards to ethical, social and ecological aspects, among other things). During this stage, we might delimit and focus our research question.

2. Starting off the project with extensive research. Both on what reasons and specific problems that causes non-diegetic GUI to break presence in VR; and making a list of what methods and design patterns are used today in VR-games, and in non-VR games, to prevent this.

3. A research plan that specifies how we will test our chosen methods, and how the data should be analyzed, so that each one method, or combination of methods, is tested in the same way, to make our results as valid as possible. It's important that we also consider ethical and social aspects at this point.

4. Come up with a good scenario and setting that works with non-diegetic GUI in VR that needs high sense of presence

5. Create a first prototype (named A) that shows the problems we've found that causes non-diegetic GUI to break presence in VR (using the scenario from previous point). This is to test against the methods we later use to have a baseline comparison.

6. Decide on what methods that we want to try out
7. Create prototypes (named B, C, etc) where each of the decided on methods are applied to prototype A
8. Test each prototype B, C, etc. on their own with testers, in comparison to prototype A, to see how well each method works
9. Analyze test data, and how well each method performs (Point 10-13: if there are more than one)
10. Decide which of the tested methods B, C, etc. to move on with
11. Create prototypes (named BC, CD, BCD, etc.) to test the optimal combination of several methods
12. Test each prototype BC, CD, etc. with testers, to find and prove the optimal combination of several methods
13. Analyze test data, and interaction effects between methods

5.1.2 Basic time plan

- First month (covering January and February):
  Researching problems and methods, and writing our planning report.
- Second half of February:
  Develop prototype A
- First half of March:
  Develop prototypes B, C, etc.
- Second half of March:
  Test prototypes B, C, etc.
  Analyze data
- First half of April:
  Develop prototypes BC, CD, etc.
- Second half of April:
  Test BC, CD, etc.
  Analyze data

5.2 Refined plans

These plans are the result of the pre-study of this project.

The planned structure of this project can generally be divided into four parts or phases: pre-study, development, playtests and analysis.

5.2.1 Planned result

- A summary of factors that affects sense of presence in VR, either by increasing or decreasing it
- A prototype “A” that clearly shows a problematic way of implementing a tutorial in VR, in regards to sense of presence
- Two prototypes “B” & “C” that improve on prototype A, showing different ways of implementing tutorials in VR that keeps a better sense of presence
- A qualitative analysis of the collected data from playtests of the three prototypes, that shows what works best for tutorials in VR

5.2.2 Plan of process

The process of the project have been divided into phases roughly four or five weeks each described in the time plan (see Section 5.2.3). The plan is to continuously write on the report throughout the project but during the last four weeks it is the only thing we will do.

Below is a more detailed description of the phases development, playtests and analysis. The pre-study phase is not covered here, since it already has happened, but the result of which is instead covered in Sections 2, 3 and 4.

5.2.2.1 Planning development

We plan on having an iterative development process (see Section 4.2 for more details).

First we ideate on what different puzzles we need to create. These need to be hard enough to warrant a tutorial, but still simple so that they can be short enough to have time with three of these in succession, for one tester.

The second part of the development will be designing these puzzles and quickly prototyping them to see if they work and how they can be improved.

When the puzzles work well enough for our purpose we continue with developing the environment and puzzles in VR.

Then we continue with the design and development of the tutorials, which should be three different versions according to the different factors of presence that should be tested.

Finally, the prototypes and playtests will be pilot tested.

5.2.2.2 Planned playtest

Each playtest will have the following layout: first we start with the introduction of VR to the tester. The tester’s experience level will be assessed, and if needed, we will explain how the controllers work. They will then get to sign a consent form about us collecting data when the participant is playing and filming them during this. The recording is not vital to the research, so if a participant don't want to be recorded, they can still participate.

During the playtest, the tester will play through three different prototypes, each with a new puzzle and a different tutorial for each. To block for order bias, we will also rotate which of the tutorials that the tester gets to play first. We also plan on rotating the puzzles among the different tutorials, so that we avoid correlations between test results for a certain tutorial and a certain puzzle.

After each tutorial the tester is asked to fill in a questionnaire about how much presence they felt (IPQ), before proceeding to test the next tutorial in order.

When the three tutorials have been tested and the three questionnaires have been answered, we continue with a short interview with the tester to try and understand why they felt more or less presence in the different tutorials. These interviews will have a general structure, but should be based on the observations and questionnaire answers during the preceeding playtest, as well.

Finally, we will thank the tester for their participation in this project, offer to answer any lingering questions, and send them on their way.

5.2.2.3 Planned analysis

After each playtest has been concluded, we will transcribe the interview, and then code the transcript and our observations. A *Deductive Content Analysis* (see Section 4.4 for more details) will be performed continuously and iteratively after each playtest, as more and more data is collected.

When all the playtests are done and the *content analysis* is finished, a wider analysis of its results will be a performed based on *Kano analysis*, where we try to define what things influenced presence and how important they were, according to the testers.

5.2.3 Time plan

Week 8 - Start of the project ideation and prototyping of puzzles

Week 9-10 - Develop the virtual environment and puzzles

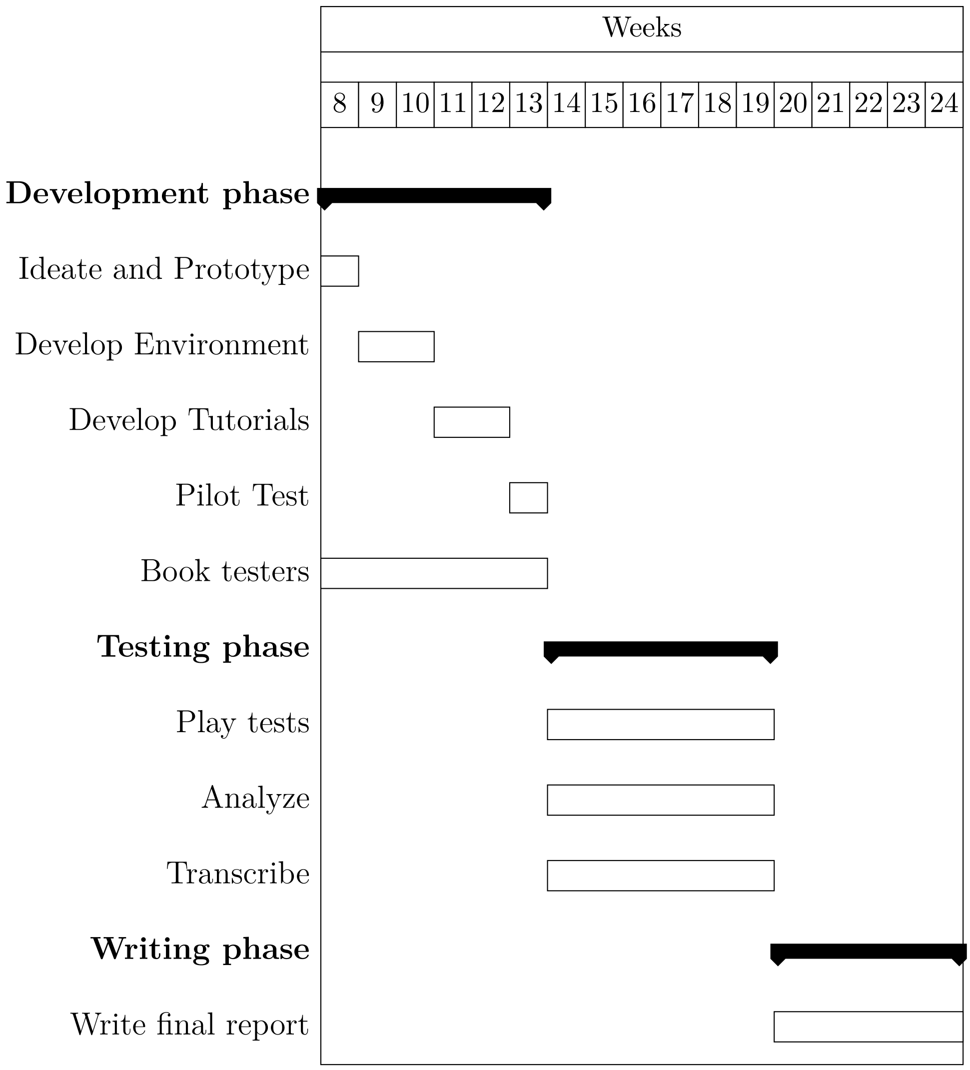
Week 11-12 - Develop the tutorials for the puzzles

Week 13 - Pilot test and bug fixes

Week 8-13 - Book testers and media lab for testing

Week 14-19 - Playtests, transcribe interviews,
and iteratively code and analyze the data

Week 20-24 - Write the report


Weeks
8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24
Development phase
Ideate and Prototype
Develop Environment
Develop Tutorials
Pilot Test
Book testers
Testing phase
Play tests
Analyze
Transcribe
Writing phase
Write final report

6
Process

This chapter details the work done in this project, following the planning, and the process is divided into the phases described in the previous Chapter 5. The development (Section 6.3) and testing (Section 6.4) phases is where most of the time went into.

6.1 Pre-Study

The pre-study was the first stage of this project and spanned four weeks. During this time, there were two main focuses: first and foremost, a basic *literature review* of the field of presence in VR was performed. Secondly, a game study was performed as well. During the pre-study, we also revised and expanded on our initial plans, as well as made a lot of important decisions for the coming phases of the project.

6.1.1 Literature review

Summing up, this covered topics such as how to measure presence specifically, what factors that might affect presence either positively or negatively, what diegesis is and how to work with it, the difference between presence and immersion or involvement, and how to design tutorials for VR. The results of the *literature review* can be found in Chapter 3.

When researching the subject, it became clear that different studies came up with different kinds of factors that might affect presence in different ways. To get a good understanding of all of these and to find out which actually would affect this project the most, we sat down and summarized all of those that we've found. First, all of the factors mentioned in different papers were condensed into manageable items and listed. We then used an *Affinity Diagram* (see Figure 6.1) to connect and group different factors into larger categories, which then were named as new factors. See Section 3.1.2 for the entire list. This list could then be used to easily see what factors we might be able to control when specifically designing a heavy interface such as a tutorial in VR, and which might not be controllable but still could affect our results, and so on.

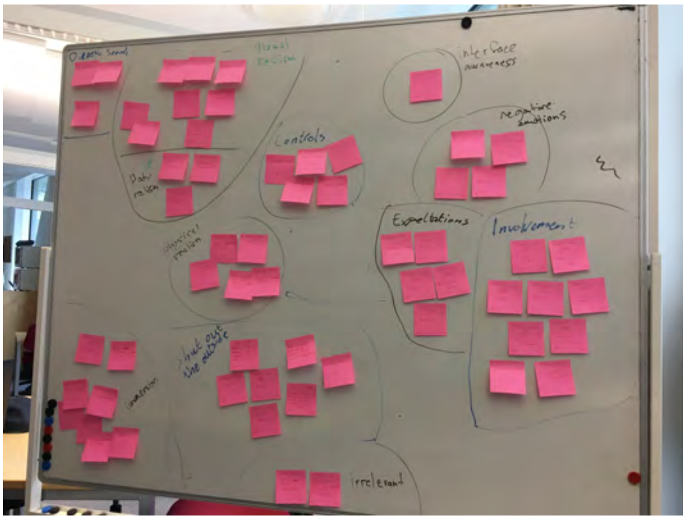


Figure 6.1: An *Affinity Diagram* based on different factors that affect presence in virtual environments, according to the results of the pre-study.

6.1.2 VR game review

In the pre-study for the project, a number of VR games were played with the intention of both getting comfortable with how VR games usually are designed, while at the same time analyzing how well each game worked with GUIs and the concept of presence, by observing each other playing the games and looking for elements that could be breaking presence. This did not follow any specific methodology, although it's vaguely similar to a fusion of *deductive content analysis*[31, p.40] and *casual observation*[31, p.120]. VR Games that we thought were relevant to the project as well as games suggested to us by our supervisor and our advisor (listed below) were played. These were at the same time analyzed (see Appendix A) with regard to the subject of this thesis, which is mainly presence and GUI. Since we are two researchers, there were constantly the two perspectives of experiencing the VR game first-hand, and observing the player from the outside. The purpose of the game study was both to get familiar with how VR games work with heavy interfaces, as well as getting inspired for the development of our own prototypes.

Included games in the review:

- Beat Saber
- Apex Construct
- The Curious Tale of the Stolen Pets
- Budget Cuts 2
- Google Earth VR
- Keep Talking and Nobody Explodes
- Obduction

Figure 6.2: Teleportation arc and using tools in *Budget Cuts 2*[10]

- Boneworks
- No Man's Sky
- Until You Fall
- Race the Sun
- Creed: Rise to Glory

Following is a summary of our thoughts about good or less good practices that was found during this study. It's crucial to point out that anything that lacks a reference purely consist of subjective thoughts and opinions that came up while playing the games or talking about them afterward.

An aspect that was found to help presence in different ways, was when interfaces was largely more diegetic than not, as for example in *The Curious Tale of the Stolen Pets*[7] or *Budget Cuts 2*[10]. When interfaces weren't directly diegetic, it still helped if used very sparingly instead, as in *Boneworks*[9]. As another interesting example, in *Creed: Rise to Glory*[6] the stamina of your character is displayed as flashing boxing gloves when you get tired. This is diegetic, in the sense that the gloves are present in the world, but non-diegetic in the sense that the flashing is not, and is not something that the other boxer can perceive.

Another thing that was experienced as presence inducing, was a high or stressful tempo or otherwise very engaging visuals, as in *Beat Saber*[19] or *Keep Talking and Nobody Explodes*[60]. It's also important to remember that audio can have a great impact, as it has in the game *Obduction*[61].

Figure 6.3: A lot of non-diegetic information in *Until You Fall*[8]

On the other side, a general lack of polish in a game, which rather might be interpreted as a lack of realism, was experienced to decrease the sense of presence in a game. Comparing *Apex Construct* with *Budget Cuts 2*, which are mechanically very similar, there was something about the higher fidelity and smoother nature of *Budget Cuts 2* that gave a stronger presence than the other. Although *Race the Sun*[62] were perceived as generally unpolished and crude, especially in the menus and interfaces, we did experience that there were less problems with the incredibly fast forward motion than expected. This was probably because the speed almost made it feel like the objects around you traveled towards you while sitting still, instead of the opposite.

There is also the problem of locomotion: while teleporting around (as in *Apex Construct*[5] or *Budget Cuts 2*) may feel less nauseating than “walking” around as in non-VR FPS-games, the act of choosing a space and teleporting there is not intuitive and might not be good for presence. On the other hand, continuous movement (as in *Boneworks*) is often very nauseating for most players. Helpful “comfort modes”, where the visual field is narrowed down to reduce cybersickness (as in *Google Earth VR*[20] or *Until You Fall*[8]), might help but at the same time also reduce presence. These ways of reducing the nauseating feelings can often become less of an issue as the player becomes more experienced and starts to forget about them. Some games have chosen to implement both ways of moving, such as *Obduction*, *No Man’s Sky*[63] or *Until You Fall*; which perhaps is a good way of at least letting players choose what works best, for themselves.

An interesting example is *Until You Fall*, which was very pleasant to play (almost no cybersickness) while still making use of both continuous locomotion and teleportation. The gameplay was hectic and engaging, and used a lot of diegetic or half-diegetic user interfaces in a good way. The game was experienced as very enjoyable, although with a low sense of presence. A guess is that the cluttered interfaces, combined with the repetition of the rogue-like genre and the arcade-like feel of the game, dragged down the presence, but not the experience. This goes to show both that presence is not the same as enjoying a game, and that enforcing presence is a very complicated task.

6.1.3 Revising the plan

During the time of the pre-study, the original plan of the project (see Section 5.1) was revised and fine-tuned to better fit our better understanding of the field. First off, we decided to construct three different tutorials, using the aforementioned list of distilled factors (see Section 3.1.2) to design each one. One of these tutorials would act as a base case (named Tutorial A), and not have anything that might enforce presence in a user. The other two (named Tutorial B and C) would then improve on the base case in different ways, according to our list of factors. Mainly, the level of diegesis were different in all three tutorials, but we also tried out different ways of utilizing depth and sound.

Then there is the question of what the planned tutorials should actually teach a user. To keep it as simple and controlled as possible, we settled on having a short but challenging puzzle for the user to play through. Since there are three tutorials to test, however, and available test users are not trivial to come by, it would be optimal for each test to cover all of the planned tutorials. Since a user cannot unlearn something they've already learned in a previous tutorial, it was decided to develop three puzzles, one for each tutorial.

Following the more refined plans on how to build the prototypes of tutorials and puzzles, the plan for testing them also had to be revised and expanded. To mitigate possible bias in the tests originating from the order or combination of tutorials and puzzles, we decided to randomize both of these things for each test user, while also making sure that each permutation (see Table 6.1) was visited an equal amount of times.

	Tutorial A	Tutorial B	Tutorial C
Puzzle 1	1A	1B	1C
Puzzle 2	2A	2B	2C
Puzzle 3	3A	3B	3C

Table 6.1: Permutations of tutorials and puzzles.

Additionally, it was decided to record video of each participant during the tests, to use as basis when discussing observations afterward, and to record audio during an interview after the participant has played through all of the three puzzles, for transcribing and coding later. The observations and interview will then be the qualitative backbone of our analysis. At the same time, in our research on ways to measure presence, a couple of different kinds of quantitative questionnaires were mentioned and used repeatedly in different studies, so we decided to also incorporate one of these questionnaires: the *Igroup Presence Questionnaire* (see Section 4.3.2 for more details on that). This questionnaire will then be answered by each participant three times, one time for each of the puzzles they play through.

We discussed different ways of analyzing the resulting data, and decided on first using a *deductive content analysis*. After this, we concluded that a custom version of *Kano Analysis* seemed like a good approach, where the given scales were instead replaced with our own deducted presence factors.

The new and revised plan that this pre-study resulted in can be viewed in its entirety in Section 5.2.

6.1.4 Further plans and decisions

At the end of the pre-study, we discussed practical issues with our supervisor. The use of an on-campus media lab, that included several computers and HMDs, was thus permitted to us, for use during both development and user testing.

We also took some general decisions at this time, such as to develop the prototypes in *Unity 3D* as opposed to *Unreal Engine 4*, and to keep the puzzles in a sitting position, both to avoid invoking cybersickness in the test users and ourselves, as well as to make development smoother. Since we got to borrow an *Oculus Rift S*, a top-of-the-line HMD, from Fast Travel Games, we were happy to use this as our main VR platform and did not spend much energy on considering the alternatives.

Finally, we made sure to send out an interest form to potential testers before moving into the development phase, in order to be able to schedule our play tests well in advance.

6.2 Design

This section follows how the methods were applied from Chapter 4 and what design decisions were made before and during the development of the prototypes.

6.2.1 Puzzles

Through the ideation of the puzzle designs, we ended up with cube based puzzles where the player needs to pick up and look at different sides of the cubes, to take advantage of VR and using the hands as interactions. We sketched these on a whiteboard to help explain to each other how the puzzle would work and look (see Figure 6.4).

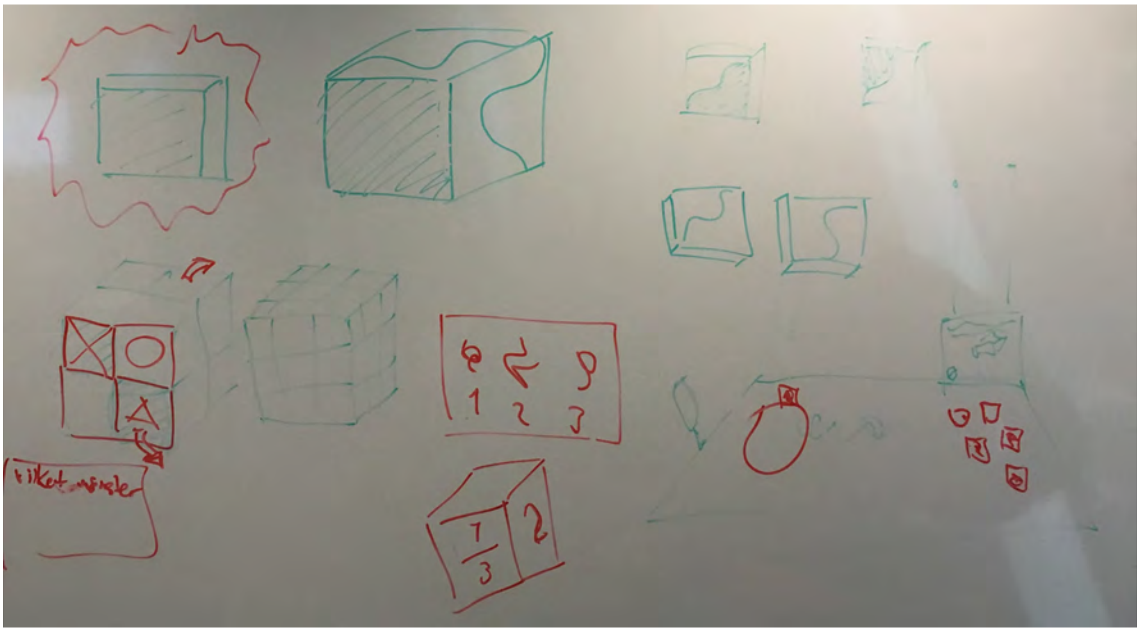


Figure 6.4: Whiteboard sketches of the different puzzles

We made fast and simple physical prototypes (see Figure 6.5) of the puzzles in order to get a feel for them before actually implementing any of them. The interactions in VR are closer to how you use your hands and head in real life, and even though you are using a controller with buttons you also move your hands and head, so the interactions with the physical puzzles matches well.

With inspiration from different standard IQ-tests, we decided on three different content themes for the puzzles, which was: recognizing patterns and their orientation; memory and sequencing; and orientation in 3D space. From these themes we thought of three different puzzles:

Puzzle 1 was about recognizing patterns. The puzzle cube therefore has a pattern on each of its sides. Those patterns are connected between the different sides to form a picture over the entire puzzle cube. One side is missing, however, and the player needs to figure out which pattern out of a collection of patterned tiles in front of them, that fits the picture of the cube.

Puzzle 2 was about spatial orientation and patterns. One famous real world example of this that we thought of is Rubik's cube, but our twist is that the player only need to create a pattern on any one side. To make it easier, we decided to only use three colors and only having a 2x2x2 sized cube. To make it more accessible, we introduced an icon for each color, as well.

Puzzle 3 is based on an idea about memory and sequence. To not make it too complicated, we decided to make the player "bump" the sides of the puzzle cube on the table in a certain sequence to solve the puzzle. So each side of the cube has two markings that they need to memorize: the first marking represents in which order and the second marking is for the number of times,

Figure 6.5: Paper and styrofoam prototyping.

that they need to "bump" the cube on the table.

An environmental theme that could fit these puzzles was a temple or cave style, so that is where we began to look for free assets. Eventually, we found enough free assets in a general style that we both could make an environment out of and that was a style that was possible to fit when creating our own assets. So we decided on a low-poly (low polygon art style) old temple style.

To finish each puzzle, the player would put the puzzle cube in a slot in the table, to see if they got it or not. The design of the table was intended to in some way indicate progress of the puzzle so that the player can get feedback of what they are doing, like a hatch that would open over the slot once the player have solved the puzzle or lamps that indicate how well they are doing. When discussing it further it only made sense to have this in Puzzle 3 and skipped it for the other two puzzles.

The design was iterated upon during the development of the prototypes and when testing it ourselves in VR. We wanted it to look like it was possible to put the cube inside the table, so it had to be thick enough to fit a cube, but we got the feeling that it could make it feel weird for the player that there is not enough room for their legs when sitting on the chair. So we decided to not have the cube fully sink into the table and could thus make the table top not as thick.

The puzzles also needed to show the win- and fail-conditions somehow. The conditions themselves are part of each separate puzzle, of course, but we decided that every puzzle will end with the player placing the puzzle cube in the slot in the table.

If the solution is wrong, a loud audio signal will sound, together with a red flashing light from the slot. If the solution is correct, on the other hand, the light will be green and the signal will be a fanfare, with the addition of confetti falling down over the player 7.12.

The cube's shape and design was inspired from a die, so that the cube looks more interesting with slanted edges and corners rather than sharp and unnatural edges.

For Puzzle 1, the slot in the puzzle cube for the tiles and the tiles themselves would have to look like they fit each other but also that it would be possible to remove tiles from the slot as well. The symbols on the sides needed to be big enough and distinct enough from each other so that the player could easily tell them apart.

For Puzzle 2, the cube was made out of 4 smaller cubes and had symbols on each side for them. First we tried adding a square, circle and a triangle, but the triangle would rotate with the cubes so it could sometimes appear to be upside down. Therefore, we instead introduced a star that looked the same no matter the rotation.

In Puzzle 3 we wanted the symbols to be carved into the cube rather than just a texture, to give depth to them. We also wanted some kind of different coloring to more easily tell them apart. Additionally, we decided that it was necessary to somehow show the progress in the puzzle, since the puzzle has such a long sequence of actions, which led to the inclusion of indicator lights.

6.2.2 Tutorials

When designing the tutorials, we looked at the derived presence factors we got from the affinity diagram (see Section 3.1.2) and picked those that we could control ourselves in the prototype. So aspects like what HMD and tracking the player would be using is not things that we can decide in the design, while the depth, sound and physical realism are things that we can control.

Tutorial A we wanted to be as close to a regular tutorial screen common in non-VR video games, as possible. This led to it being flat and having a look of an abstract screen without any presence in the room, without any sounds, physics or depth.

Tutorial B was planned to be an improvement of Tutorial A, so we wanted to apply some of the aforementioned presence factors without making it completely diegetic. To keep it from being diegetic we still wanted the look of a screen and something that doesn't fit within the world itself and is obvious that only the player can see. We used the design of Tutorial A as a template and added depth, sound and visual realism. Instead of flat text we also wanted three-dimensional text and objects, rather than just pictures.

Tutorial C was made to be as diegetic as possible. We brainstormed different diegetic visuals for the tutorial that could show different screens. First we thought of a book that had pages with the instructions on, which evolved into a pop-up book that would make it look like it could be in the world. Then

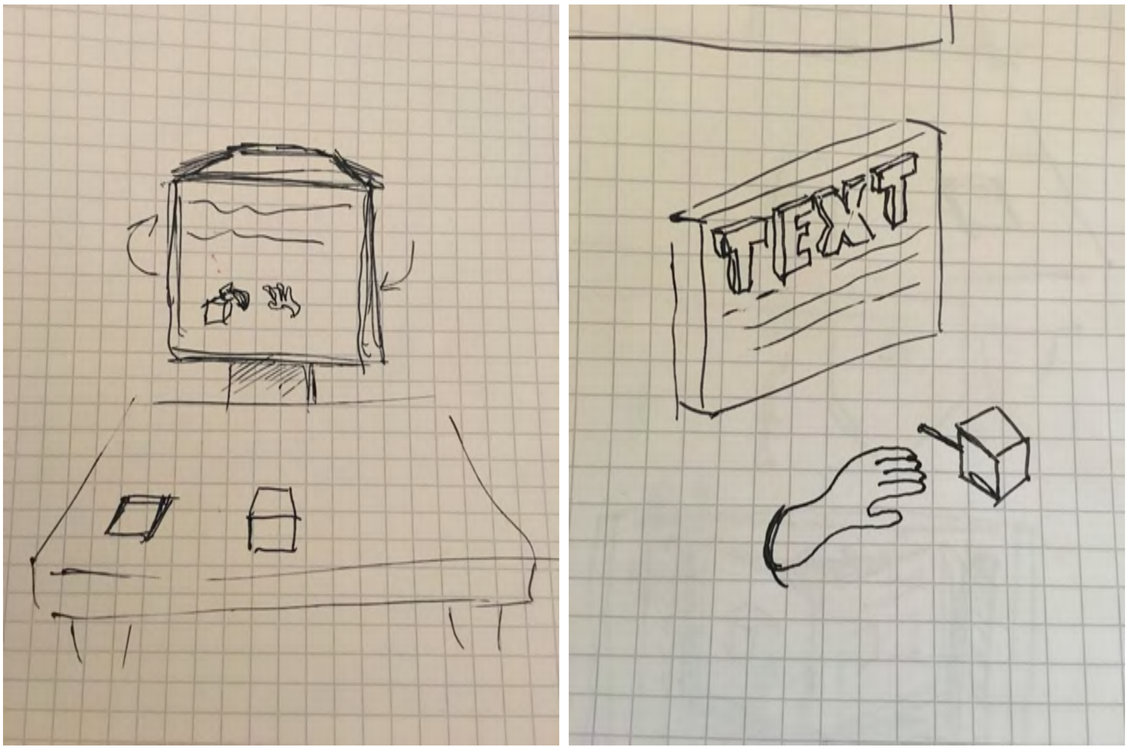


Figure 6.6: To the left: a sketch of Tutorial C, and to the right: a sketch of Tutorial B.

we thought that, since every puzzle is based on cubes, we could use a cube to display the tutorial as well, because it's a familiar object in the world. We decided on this, and that the cube could rotate to show different its sides with tutorial slides on.

6.2.3 Environment

When designing the environment we started by ideating and sketching on a white-board on how it would look like with the player in the center of the room and a skylight to light up the environment and the play area. We also wanted a table in the middle with some kind of slot to put the cube in when the puzzle was solved. In the room we wanted some holes, doors or windows so that it would look like there are other stuff around where the player is and not a just a sterile play environment.

Part of the environmental design also took place when modeling it out of assets during development. This was approached with the previous notions in mind. First the central room that would contain the play area was created, with the idea of a circular room with a lot of different archways leading out from it, in mind. This ended up having eight walls, four of which had doors leading off, and four of which had windows, niches or other details. One of the doors leading off was closed, but the other ones all got their unique design. Since the idea was that this game took place in an abandoned temple, every room visible from the central one was designed to be broken or somehow caved in. Lastly, the upper part of the central room was

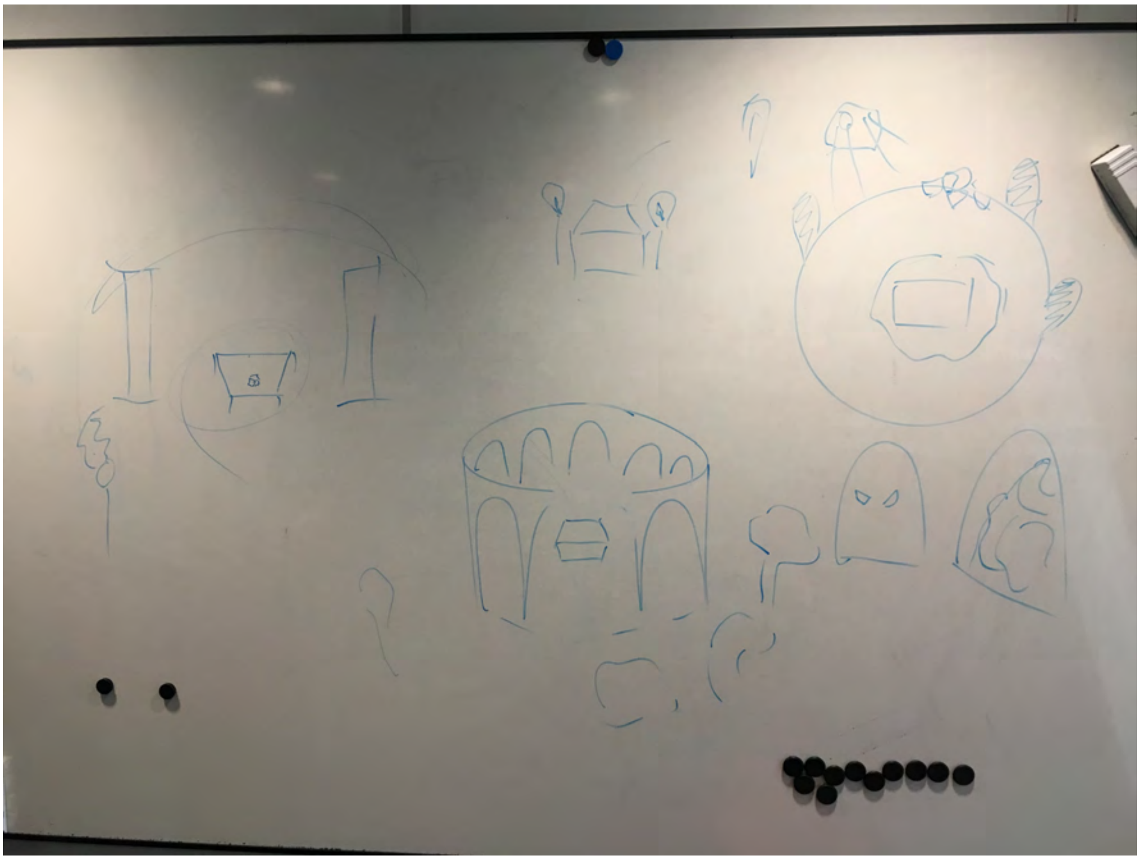


Figure 6.7: Ideating the environment on a whiteboard.

designed to be tall and domed, but with a rocky tunnel leading up to the "surface", where the sun came down to shine on the player and the playing area.

From the presence factors (see Section 3.1.2) we derived that unnatural sounds or music that the player don't expect to be in the environment was important to avoid. Thus we decided to not use music but rather spatial ambience to block out the sounds from the real world. The room ambience was combined by different types of ambience sounds from outdoor nature, as well as wind sounds. These sounds were positioned at different places in the room with some reverb, to make the ambience feel natural in VR. There were also impact sounds added for all of the objects that one can move around in the world, as well as sound effects for the tutorials, scanner and win-slot.

6.3 Development

The development of the prototypes that later will be tested, can roughly be divided into: the environment setup of everything peripheral that was needed for the puzzles and tutorials to work; building the puzzles and making sure they were functioning as planned; building the tutorials based on the puzzles and our research; and then having to revise our plans on account of the Covid-19 pandemic, before making the finishing touches.

6.3.1 Environment setup

As in most software development projects, development begun with setting up the project files and development environment. First and foremost, this entailed setting up a *Unity 3D* project and starting up a *GitHub* repository, but also included things like making sure that a programming IDE has the correct dependencies installed. Part of setting up a *Unity* project in this case also means installing a good range of VR packages in order for development to start, including the *Oculus SDK*. As beginners to VR in *Unity*, some research was needed to know where to start, and what to include or not in the end.

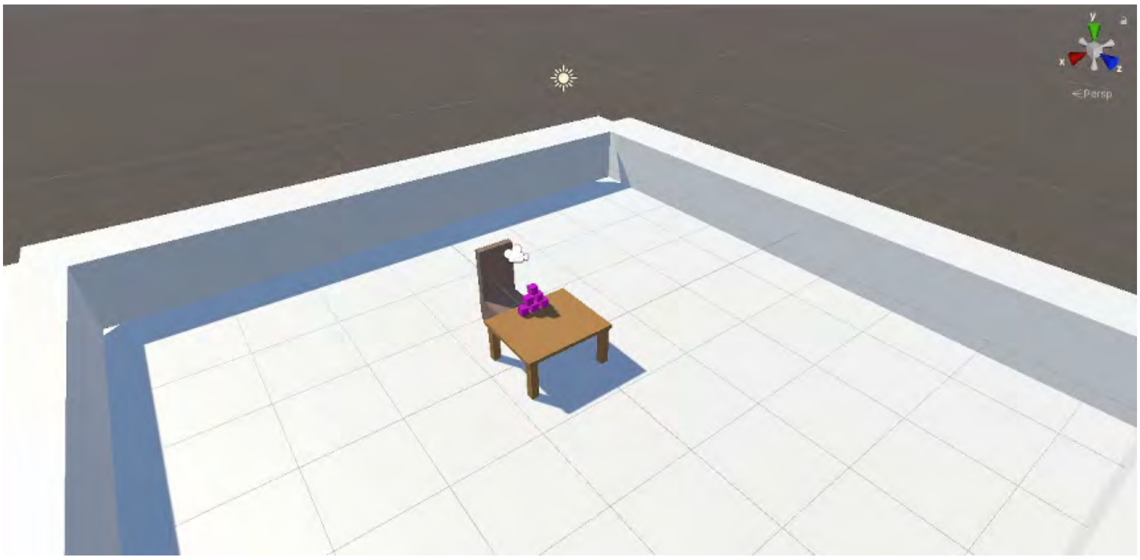


Figure 6.8: An early development environment.

We then started working with general interactions between the hands, objects and the world. This included actions like picking up an object, transferring an object between hands, and re-spawning an object in front of the player on the table, if a player places or drops it where they can't reach it. The interaction between hands and objects were based heavily on an *Oculus VR* package that we installed when setting up the project, but with added tweaks and functionality to better fit our specific goals. The re-spawning of objects was built from scratch by us, but had to be adapted and adjusted several times during development, when the level of complexity increased in the objects to re-spawn.

Simultaneously, we looked around for free *Unity* assets (mainly 3D models and materials) that could work together in a coherent theme while fitting our design thoughts and plans for the environment. We were also planning on making our own assets from scratch, but that takes an incredible amount of time in comparison, so the more free, pre-made assets we could find beforehand, the better. Starting out with this also let us match the artistic theme in those few assets we still had to create ourselves, with the free assets we'd found.

During this first phase of the development, the groundwork for the environment that the puzzles would be placed in was also laid out. Quick and easy models was built out of standard geometric shapes to get a feel for the scale and size of important game elements, such as the chair that the player would be sitting on, the table in front of the chair that the puzzles would be placed on, and the room that all of this would exist in. Getting this right was important for later, when players are put in the finished environment, because if the scale of something felt off in any way, this could dramatically affect the amount of presence felt during the tests, and therefore possibly the whole study.

When the environment started to take form on a more artistic level, with thematic props and colorful surroundings, a layer of ambient spatial and ambisonic sound was added as well. This is an important factor for presence, both to make an environment feel natural, and to block out the real world. We spent some time to make sure the spatial aspect of this worked well, and that the combined volume of all sources was not too much, but overall, this did not take too much time of the development.

6.3.2 Puzzles

When the environment for the prototypes started to fall into place, we began implementing the design for each puzzle. Part of this was to create 3D models in *Blender*[64] of the table that the puzzles would take place on, as well as the cube that each puzzle would be based on. When a generic cube design was in place, variations of it was re-modelled to fit the specific puzzles, such as creating an indented slot for the tiles in Puzzle 1, or carvings of different symbols in Puzzle 3. Additionally, some extra models were needed, for example the tiles for Puzzle 1 or the scanner for Puzzle 3.

The base mechanics of Puzzle 1 are pretty straight forward: the player should be able to place one of several separate tiles into the indentation on the surface of a cube (see in Figure 6.9), so implementing this was not very difficult. However, some fine tuning was needed for when removing a tile from the cube again, since there was a great overlap of collision volumes when trying to grab the cube or the tile separately. Later, when applying a texture of pattern that would encompass the whole cube (except the side where the tile would be attached), it appeared to be a great challenge to get the UV-mapping of the custom-made cube to work as intended, in combination with the separate tile patterns, without graphical artefacts.

Perhaps the most advanced puzzle mechanic to develop was that of Puzzle 2, where a composite cube of 8 smaller cubes were to behave as a Rubik's Cube[65]. This problem was two-sided: on the one hand, we needed to create a system for keeping track of all the smaller cubes while rotating them correctly and with animations. On the other hand, the player would need a simple way of interacting with the cube and choosing both which side of the cube to rotate, and in what direction.

Our first thought was to let the user grab the cube with both hands and then counter-rotate the hands, but this proved to be hard to implement, since we at the same time wanted the player to be able to easily pass the puzzle cube between their hands. Instead, we worked out a way of pointing at the cube with the off-hand,

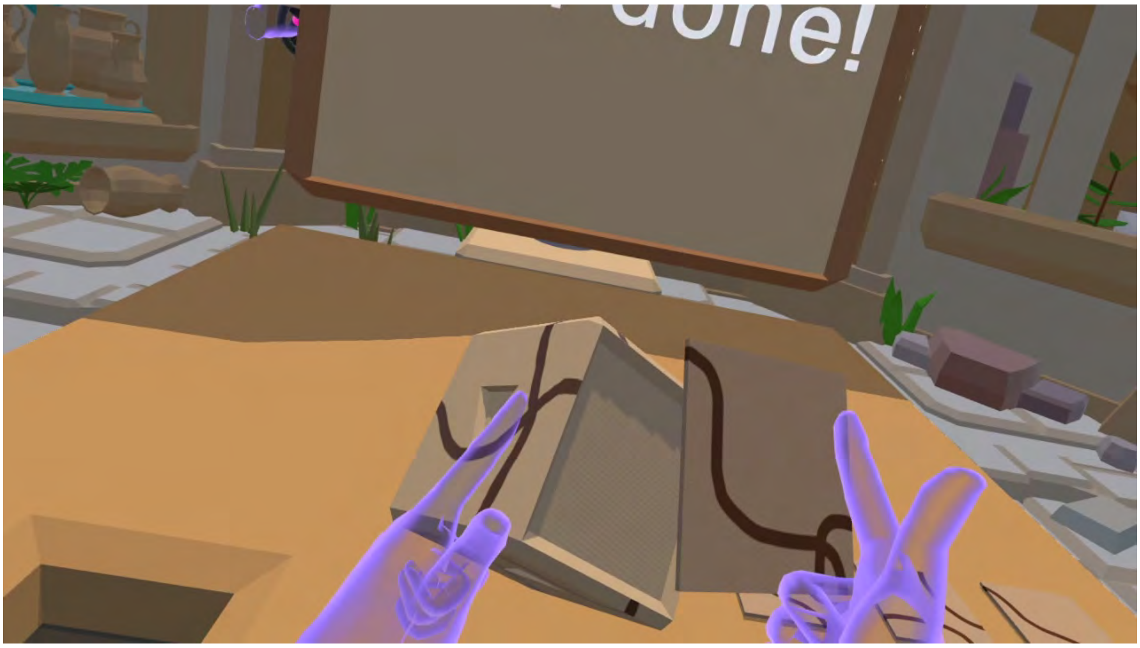


Figure 6.9: Puzzle 1 with attachable tile.

highlighting the side that would be rotated and showing an arrow for direction, like in Figure 6.10, before pressing a button on the VR controller to initiate the rotation. Even though this solution worked, it was regrettably a step away from the otherwise rather intuitive puzzle design, which was indeed noticed by playtesters later (see Section 6.4). Getting the puzzle mechanics for Puzzle 2 to work took by far the most time among the puzzles, during development.

The original design of Puzzle 3 included a way of “bumping” different sides of the puzzle cube against a scanner of sorts. When starting to implement this, though, it became clear that the impact-part of it was much more complicated than the scanning part, and when thinking about it, it was actually redundant. So the impact was removed, and thus the puzzle was distilled to just hovering the correct side of the puzzle cube above the scanner to register it (see Figure 6.11).

Another idea that was removed during development of Puzzle 3, was to have some sort of hatch above the winning slot that only opened when all the sides of the cube had been scanned in correct sequence. Removing this seemed natural when working on it, as we realized that just registering a fail and resetting the scanner sequence lights was simpler and just as easy to understand. However, the functionality of the hatch in question had already been implemented by the time this was realized.

Part of the puzzles were also a win- or fail-condition, in the form of a slot in the table that reacts with sound and light, depending on if the player wins or not (see Figure 6.12). None of this was especially hard to implement, but the light was added quite late during development, since a lot of other functionality had taken precedent before that.

Besides the sound of winning or failing a puzzle, impact sounds were added and

Figure 6.10: Puzzle 2 with the interface hints before rotating.

Figure 6.11: Scanner and cube in Puzzle 3.

Figure 6.12: Left: the wrong solution and blinking red light. Right: the correct solution and a blinking green light.

automated for every movable object, so that throwing around a puzzle cube would have an audible effect. Some energy was spent on finding a collection of sounds that was both different from each other while not *too* different to feel off. These sounds also had to be edited and cleaned up quite a bit, to not break the illusion, but admittedly, more work could still have been spent on this for better results.

6.3.3 Tutorials

As mentioned in Section 6.3.4, we would need to create a range of different permutations of puzzles and tutorials for the user testing later on, to avoid bias. So when starting to work on the tutorials, it was soon clear that it would be ineffective to manually add the same kind of tutorial texts for each and every permutation that included it. Instead, we worked out a general solution with abstract classes for tutorials, that all linked to the same text files, that could in turn be edited and changed in just that one location, and then distributed automatically when building the different executable test files.

To map out what tutorial steps that were needed for each puzzle, we waited to decide this until the puzzles were more or less finished. At that point, it was simple to list all the necessary steps for a player to win each puzzle, because we could just play through them ourselves and try it out.

During this mapping, partially because of the readjustments needed on account of the current pandemic (see Section 6.3.4 for more details on that), it also became clear that some kind of on-boarding tutorial was needed for each user tester. An on-boarding tutorial would in this case make sure that a tester was able to grasp the basic interaction of the world (such as grabbing objects, switching hands and respawning objects in front of the player), before diving into the more complex puzzles. However, it would not make sense to have the same on-boarding tutorial at the start of each test, if a tester would do three of them in a row. We therefore decided to dynamically make sure that extra on-boarding tutorial-steps were added *only* to the first of the three that each user tester would go through.

The visual style of Tutorial B (see Figure 6.13) was planned to utilize three-dimensional depth but not seem like it realistically could fit into the designed environment of

Figure 6.13: First instruction screen for Puzzle 1 in Tutorial B.

a temple ruin. Parts of the frame for it was therefore modelled in *Blender*, combined with simple geometrical 3D objects in *Unity*, and colored with a neon-like pearlescent color palette that clearly stood out from the temple stones strewn about the room. To further accentuate the depth of the tutorial, the text and descriptive graphics for each tutorial-slide was also rendered in 3D, and animated to rise in and out of the tutorial screen. As another measure to grab the user's attention, an abstract science-fiction sound was added to the transitions between slides, which was not supposed to fit with the environment, either.

As we settled for the idea of a big rotating cube for Tutorial C (see Figure 6.14), we could use the same 3D model that were used for the puzzle cubes. Later on, specific frames had to be modelled as well, though, to make each slide of the tutorial cube clearer.

An interesting aspect of the rotation of the cube was that for each puzzle, there were more than four slides to display, which means that the four sides of the cube did not suffice. Therefore, a mechanic of spawning and de-spawning slides on the back of the cube was implemented, thus making the possible amount of slides infinite, without drawing attention to the mechanics behind it.

Just like in Tutorial B, a transition sound was added, but in this case, it was a sound of scraping stones, in order to work better with the rotating cube, as well as the environment. The text and graphics for Tutorial B was completely reused in Tutorial C, which thankfully reduced the amount of time that we had to spend on that. Since Tutorial C was designed to also be physically present in the environment, both the rotating cube, the text and all the graphics had to be given physical colliders, so that if a player would throw an object at it, it would bounce off instead of go right through it.

Figure 6.14: First instruction screen for Puzzle 1 in Tutorial C.

Figure 6.15: First instruction screen for Puzzle 1 in Tutorial A.

While the frame for Tutorial A (see Figure 6.15) was the first to be implemented, it went fast because it was designed to just be a flat image hanging in the air. The text was just as easy to implement, since there are tools for dynamically rendering 2D text in *Unity*.

However, we waited to implement the graphics of Tutorial A, because we could then use the three-dimensional ones from Tutorial B and C, and simply take screenshots of them when viewing them through an orthographic camera. This way, we did not have to create new flat graphics for each slide again. On the other hand, since the conversion from 3D to 2D was manual, we had to manually repeat the conversion every time an unplanned change occurred anywhere in the graphics. It's still rather clear that we saved a lot of time on this approach, though.

6.3.4 Covid-19 pandemic and revisions to the plan

About halfway through development, the global Covid-19 pandemic reached Gothenburg, and the on-campus media lab in which the development took place had to close down. This threw us into a tumultuous couple of weeks where the nature of the development, testing and entire project was forced to change. All the remaining plans of the project had to be overlooked, considered and adjusted, to make sure that the project could continue and would be possible to finish.

In the end, this lead to us borrowing equipment from the media lab and moving development to our respective homes instead. So in general, development could continue more or less as before, although we did not have time to do much actual work during those couple of weeks, on account of the readjustment and movement of equipment, and so on. We therefore decided to extend the development phase a week, to make up for lost time, and also to have time to find new testers.

There were a couple of larger impacts on the development, however. Firstly, since we couldn't be present during the tests anymore (because of the pandemic), it was much more important to make sure that each tester actually understood the basic interaction of our environment. This was part of the reason that we added the onboarding tutorial parts to the first test file in each test sequence (described in more detail in Section 6.3.3).

Secondly, we no longer had the same opportunity to pilot-test the prototypes. Pilot testing was an important step at the end of development that was intended to make sure that no obvious problems or bugs were present in the prototypes when actually testing them in the study. Since available testers that had their own HMD were much more scarce than those without, we had to skip the pilot-testing entirely. We did not know for sure at the time, but we suspected that this would have an impact on the finished testing product, a concern that later during testing seems to have been justified (see Section 6.4).

Finally, since the testing of the prototypes had to change to avoid personal contact, the distribution of the tests was now a factor that had to be considered, to allow for remote testing. Our main platform for distribution had up until this point only been focused on *Oculus Rift* and *Oculus Rift S*, which were the two HMDs we used during development. These are entirely compatible with each other, so no effort was needed to make the prototype work between them.

When finding new testers that had their own HMD, on the other hand, not everyone had *Oculus Rift*. This meant that we had to spend quite some time on making sure that we not only could build executables for *Oculus Rift*, but also for *Valve Index* and *Oculus Quest*, which was problematic since we couldn't actually try them out ourselves and see if it worked. We ended up skipping *Valve Index* completely, but managed to convert the *Unity* project to an *Oculus Quest* version of the *Android* platform. After quite a few speed bumps, this finally worked out fine, which resulted in us having several extra user testers.

Because of all the possible permutations of the executable builds between puzzles and tutorials (18 in total, see in Table 6.1), we fashioned an automatic system for building all of them as soon as new functionality was added, instead of having to build each and every one manually. This was not trivial, but saved us a lot of time in the end, especially when automating the *Android* builds as well.

6.4 Testing

The structure and nature of the tests had to change drastically because of the Covid-19 pandemic. Originally the plan (seen in Section 5.2.2.2) was to have the tests in person so that we could observe the players during the tests and help them if there would be any problems. When the university closed, we could no longer use the media lab for our tests, and the solution that came from discussions with our supervisor, was to do these tests remotely instead (see Figure 6.16).

The entire plan of the tests therefore had to change: the testers now had to be people that already owned a VR headset themselves. This lead to the number of testers going down from 24 to 16, after some intensive re-booking. So all of our new testers were more experienced VR users than previously planned, when there was a wider range of experience. Some of the new testers were developing VR games themselves, and therefore their focus and experience during the tests might be very different from the average VR user. In addition, we had very few testers with low experience in VR.

We discussed asking the testers to try and set up a camera in order to share their screen, but with the amount of extra work needed of the testers and the number of things that could be problematic, we decided not to observe them while playing. We did not want it to be hard or annoying for the participants to test our prototypes, so we decided to not require these things and let them play through the prototypes by themselves.

Because of the decision to develop with the *Oculus SDK*, it limited us to test with people that had an *Oculus* headset and thus also limited the number of testers available to us. If we instead had made the decision from the start, to develop with a more open development kit, we could possibly have gotten it to work on more devices. But we did not have the ability to test if a certain build worked on those other devices ourselves, which later was the problem with the build for *Oculus Quest*.

Figure 6.16: A user play testing Puzzle 1 with Tutorial C, at a home based working station.

From a technical standpoint, we decided to use *Hangouts*[66] for our interviews, so that the participant did not need to install new software to run it and could instead just use a web browser. Both of us each set up recording software to be able to record the audio of ourselves as well as from the call, so that we always have a backup if something would go wrong with one of the recordings. For the consent form and the questionnaire, we decided to use *Google Forms*[67] to make it easy for the testers to answer online. There was a discussion to have the questionnaire in VR, but we decided it would take too long to develop for the tests, even though it would make it easier for the testers to have everything in the same place.

To make things simple for the testers, we built individual executable files of the prototypes for everyone. Each file name was coded like this: "1_T2CO", where the first number was the order in which they should play them, the "T2" part stood for which puzzle (in this case the second puzzle), the "C" stood for which tutorial they will see (in this case Tutorial C), and finally the "O" stood for On-boarding.

On-boarding meant that the tutorial would be extended in the beginning, to teach the player the basic mechanics that are present in all three puzzles, and it was only ever included on the first puzzle they played, since they then would already have learned the basics and didn't need it for the following tests, which would just take extra time for them.

Because of the remote testing, the files and forms all had to be sent out to the testers in advance, so that they could double check to see that everything is working before they were going to test. For this we made a checklist in order to not miss anything:

- Instructions for the test
- Link to *Hangouts*
- Consent form (see Appendix C)
- *Igroup Presence Questionnaire* (see Appendix D)
- Files for tutorials A-C

At the start of the tests, we first met the tester in *Hangouts* to introduce ourselves and the study, and to see if they had any questions. We then made sure that they had signed the consent form and that they understood that the interview would be recorded, and why. Finally before they could start play-testing we told them that they should start with the file that starts with 1, and after they had played that one they would answer the questionnaire, and then they would continue with the file starting with 2 and so on.

We then let them play on their own, and if they had any questions they could just ask us, since we stayed in the *Hangouts* call the entire time. When they had answered the questionnaire for each of the puzzles, they would join us back in the call for the interview.

After they joined us, we started recording, and one of us lead the semi-structured interview while the other took notes. We had a couple of written-down questions to guide us:

- Did the testing go well? Everything worked as it should?
- In which tutorial did you feel the **most** presence? (Why?)
- In which tutorial did you feel the **least** presence? (Why?)
- Did you feel that anything specific **broke** presence, during any of the tests? (Why?)
- Did you feel that anything specific **increased** presence, during any of the tests? (Why?)
- Did you notice that every puzzle had a different kind of tutorial? Did you feel any differences between them with regards to presence?

- Was there anything else interesting that you thought of, during any of the tests?

If the user said something interesting, we would ask follow-up questions about that. This meant that the testers could better describe their experience and with their own words say what gave or took away their sense of presence. At the end of the interview we explained more about the study and what this data would be used for and where it will be public.

After each test session, we transcribed the recorded interview, word by word, so that everything they said was put down in writing. After all of that, we went over each transcribed interview again and revised them, to make it more readable without changing the meaning of the sentences. In case that would happen anyway, we saved the originals to be able to double check, when needed. The revised versions were intended to make the following analysis easier (see Section 6.5).

We started the testing with two people that had a very close connection to the project and already knew a bit more about it. Although we canceled the formal plans for pilot tests, these acted as such, and let us see if we needed to change anything in our approach for the tests and interviews. A few minor problems with the questionnaire was resolved and we also learned that it would be a good idea to explain that it's a seated experience with headphones, so that they could hear the sounds. The feedback came too late in the development process, however, to fix any problems within the actual prototypes. This could have greatly improved the general experience for the testers, and a lot of issues that broke their presence could potentially have been fixed.

The tests went according to the plan for the most part, and only during one of the tests was there a problem with the test files they had gotten, so that they could not complete all of the puzzles.

6.5 Data Analysis

Analysis of the data included a *Deductive Content Analysis*, a qualitative analysis of the results of that with focus on the derived factors (see Section 3.1.2), and a comparison between our main qualitative findings and what quantitative findings we could see.

6.5.1 Deductive content analysis

When all user tests had been conducted and every interview was transcribed and revised, we went on to perform a *Deductive Content Analysis*. The "deductive" part of it means that a set of predetermined themes were used as codes when coding each interview. We used the previously derived 11 presence factors (see Section 3.1.2) as our themes, but also added a couple of additional codes that emerged during coding.

Coding each interview meant going through each of the revised interview transcripts and selecting quotes that were relevant to our research. During coding we had this list of guidelines that we followed as best we could:

1. Focus on the presence factors as codes, first
2. Be open for additional emerging codes, though
3. Keep track of what puzzle and what tutorial each quote is about (this might mean all of them, or a combination)
4. Add a personal ID code for each interviewee, in order to distinguish between unique opinions, and so on
5. Only add new quotes from the same person, if it adds new codes or nuances to the already coded data
6. No translation of quotes at this point
7. Clarify words or show excluded sentences in the middle of a quote, with brackets

The fifth guideline "Only add new quotes from the same person, if it adds new codes or nuances to the already coded data", was decided on to not let certain interviewees that were especially verbal or repetitive, to saturate the data with a certain view or opinion.

This is also part of the reason that we kept track of each interviewee in the data with a personal ID-code. However, we used ID-codes instead of actual names, in an attempt to reduce personal bias.

We wanted to keep good track of what tutorial or puzzle, specifically, a quote referred to. This was because even if an interviewee was talking about their presence during a certain puzzle, they might not realize that the specific tutorial displayed during this puzzle might affect their opinions somehow.

The interviews were mainly conducted in Swedish, except for a couple of them that were conducted in English, instead, which meant that the data contained quotes in both languages. We decided to keep the quotes in their original language, to not change their meaning too much.

In addition to the quote ID-codes and what puzzle and tutorial they referred to, we also allowed for up to three different codes for each of them, ordered after priority. Since one often covers several subjects at once while speaking, allowing for more than one code for each quote was crucial. At the same time, however, we also constrained it to a maximum of three, in order to keep the workload at a reasonable level.

During coding, four new codes emerged:

- Difference in tutorials
- Acclimation
- Flow-state

- Difficult to understand

The code “Difference in tutorials” were given to quotes that addressed the perceived difference in tutorials, since this was a major thing that we were interested in. “Acclimation” meant getting used to the controls and the environment, which apparently had an effect on the perceived level of presence for a tester. “Flow-state” means the concept of forgetting everything else when performing an especially engaging task, which is close to, but not the same as, presence[40, 41]. “Difficult to understand” were given to quotes that covered something being hard to understand, be it the controls or the text in a tutorial.

These extra codes did not have a large impact on the analysis, though, partially because they were not used as much as the predetermined factors, and partially because either they were not concrete factors that impact presence, or they could be included under one of the other factors. They were still helpful, though, since they let us identify certain aspects or patterns that were interesting to our study, all the same.

6.5.2 Factor analysis

When planning the project, we decided to continue with a custom version of *Kano Analysis* after the *Deductive Content Analysis*, where the Kano scales were exchanged with our derived presence factors (see Section 3.1.2). However, when going into the analysis phase of the project, it suddenly became clear that we actually had misinterpreted how the *Kano Analysis* worked, and thus, we had to adapt our plan to this.

What we had planned to do, was to go through all of the coded data, one factor at a time, and see how much each factor influenced the feeling of presence in users. The misunderstanding with the *Kano Analysis* was that we could somehow combine these, which turned out to be untrue, since its scales and interpretation matrix was very much coupled with the specific scales it used. Instead, we just dropped the *Kano Analysis* and went on with the other half of the plan.

So working through and elaborating on our original half-plan, we decided to systematically read through each and every quote, focusing on one factor at a time, in priority order. We then interpreted and summarized ideas and concepts brought forward by each interviewee, and most importantly, clustered those that expressed similar ones. This way, we got a good feeling of prominent themes and what were more influential than others. The resulting factor analysis can be viewed in Section 7.3.1.

To reconnect to the purpose of this study, after going through each factor, we also summarized what our findings told us about designing tutorials for VR, specifically, while sifting through the more general thoughts on presence in VR. The results of this summary was 20 guidelines, which can be viewed in Section 7.3.2.

6.5.3 Quantitative findings

As this is mainly a qualitative study, focus was never on generating statistically reliable quantitative data. None the less, we did end up with quite a bit of quantitative data anyway, that were analyzed and compared to our qualitative findings.

After going through the *Deductive Content Analysis*, and our own Factor Analysis, there were a lot of coded quotes with puzzles and tutorials assigned to each one. We therefore drew up diagrams of the total number of codes used, in what priority order they were used, and finally also the total number of codes used in relation to the different puzzles and tutorials, respectively (see Section 7.2.2 for these diagrams). We then went through the different diagrams and compared to our qualitative analysis of the data, and compared what seemed to match and what didn't. Some of our conclusions had slight support, while others did not, and the diagrams also showed some correlations that we did not see before (however, these are not verified for statistical significance, so all of these conclusions should be taken lightly).

Apart from the analysis of interview data, there were also the *Igroup Presence Questionnaire* that each participant filled in. Initially looking through the data, there were no obvious pointers that correlated with our qualitative findings. So when compiled to general presence scores, we got a better feeling for the differences (or lack thereof) between different puzzles and tutorials. We also examined the order in which each participant tested the puzzles and tutorials (since this was randomized between the participants), because this was something we were worried could affect the results. All of this helped us to get a better understanding of the results, and showed us what testers thought was more or less important, although it did not significantly impact any of our conclusions in the end. These results can be found in Section 7.2.1.

7

Results

First, we'll show and describe how the prototype with different tutorials and puzzles ended up. Then we'll go through the quantitative and qualitative data, and finally concluding the results with a list of guidelines for designing tutorials for presence in VR.

The results are mostly qualitative, but from the questionnaires there was some quantitative data since each of the 16 testers answered the questionnaire three times each (except for one where the they could not complete the test due to technical problems) so in total 47 questionnaires answered. Since the sample size is so low, no general conclusions from these results can be made about what increases or decreases presence, but they can indicate what the testers thought were more important as well as what they focused on and what differences in the tests they noticed.

Figure 7.1: An overlook of the environment for Puzzle 1 and Tutorial C.

7.1 Prototype

In all the prototypes the same environment (see figure 7.1) was used, which was built in *Unity* with free assets from *Unity Asset Store*[68, 69, 70, 71, 72, 73, 74]. The prototypes are available to download at itch.io[1] and the project files are available at Github[2].

So the prototypes had three different puzzles and each puzzle had three kinds of tutorials (available permutations can be seen in Table 6.1). In Puzzle 1 (seen in Figure 7.2) the player needs to find what tile that fits with the patterns on the sides. They put the tile in the slot on the cube and they can swap it out with another tile if they don't have the correct tile in the correct orientation. They can then check their solution by putting the cube in the hole in the table. Besides matching the patterns on the cube, they also have to find a tile with a symbol on its back that matches the opposite side of the cube according to a given legend.

Tutorial A is the flat, slightly transparent screen with flat text, without any sounds (also seen in Figure 7.2), and between each slide of information there is a slight fade.

Figure 7.2: Puzzle 1 with Tutorial A, showing the first slide of the on-boarding parts.

Puzzle 2 (see Figure 7.3) is based on a 2x2x2 Rubik's cube. It has 3 different symbols with different colors that the player has to match with a given legend. To rotate the faces of the cube, the player points with the other hand than the one holding the cube, and an abstract interface hint will show in what direction the selected face of the cube will rotate in. In order to rotate the player presses the trigger button on the pointing hand. When the correct pattern is visible on one of the puzzle cube's sides, the player can put it in the finishing slot in the table to clear the puzzle.

Figure 7.3: Puzzle 2 with Tutorial B, showing the first slide of the on-boarding part.

Tutorial B (also seen in Figure 7.3) has more depth than Tutorial A, with three-dimensional text and infographics. When transitioning between slides, the content moves in and out of the background with a sound effect.

In Puzzle 3 (see Figure 7.4) the player has to scan the sides of a puzzle cube in a certain order and times. The Greek symbols on each side represent a number that are decipherable with the help of a given legend. For each correctly scanned side, a small indicator light will turn on, and when all six lights is on (for each of the cube's sides), the player may put the cube in the finishing slot in the table to clear the puzzle.

Tutorial C (also seen in Figure 7.4) is a large cube where each side is a slide with three-dimensional text and infographics, like in Tutorial B. The difference is that the entire cube rotates when transitioning between slides, with a sound effect of stones scraping against each other. And while Tutorial A and B does not have any physical rules applied to them, which means that you can throw objects straight through them, Tutorial C is just like any other object in the environment, and therefore will stop thrown objects just as a wall would.

Regardless of what puzzle that is active, the player always has the option to put the puzzle cube in the slot in the table. If the puzzle solution is wrong, a beep will sound and red light will shine from the slot. But if the solution is correct, green light will shine, a fanfare will sound and confetti will be shot out and over the player from behind the tutorial. The text "Congratulations! You have finished the puzzle!" will show on the tutorial screen, and that specific experience is thus over.

Figure 7.4: Puzzle 3 with Tutorial C, showing the first slide of the on-boarding part.

7.2 Quantitative data

In this section, we go through the resulting quantitative data from our study, starting with the questionnaire that each participant filled in. After that, we go through the quantitative data the analysis of the interviews resulted in, and how one might interpret that.

7.2.1 Data from questionnaires

Each test participant got to answer the *Igroup Presence Questionnaire*[28] three times in total, one for each tested prototype. The results from the questionnaires gave us a total presence score for all of the tests, that we can compare with specific presence scores for each tutorial, puzzle or in which order they were played. All of the questions could be answered on a scale from 0 to 6, and the total presence scores are then an average over these, where 6 is the highest sense of presence and 0 is the lowest sense of presence.

The *Igroup Presence Questionnaire* is also divided into different scores for the sense of the perceived user presence on four different subscales (*General Presence*, *Spatial Presence*, *Involvement* and *Experienced Realism*). One of the questionnaire questions is coded for the General Presence, while the other subscales are an average between 4-5 questions each. In Table 7.1 we can see the averages of these subscales and the average over all questions.

Interestingly, the Experienced Realism over all participants is the lowest scoring category and is the only one that has a score under 3 (as seen in Figure 7.5), which is also the furthest from the mean of them.

General Presence (PRES)	4.5
Spatial Presence (SP)	4.6
Involvement (INV)	3.9
Experienced Realism (REAL)	2.8
Presence Score	3.9

Table 7.1: Presence scores from *Igroup Presence Questionnaire* over all participants.

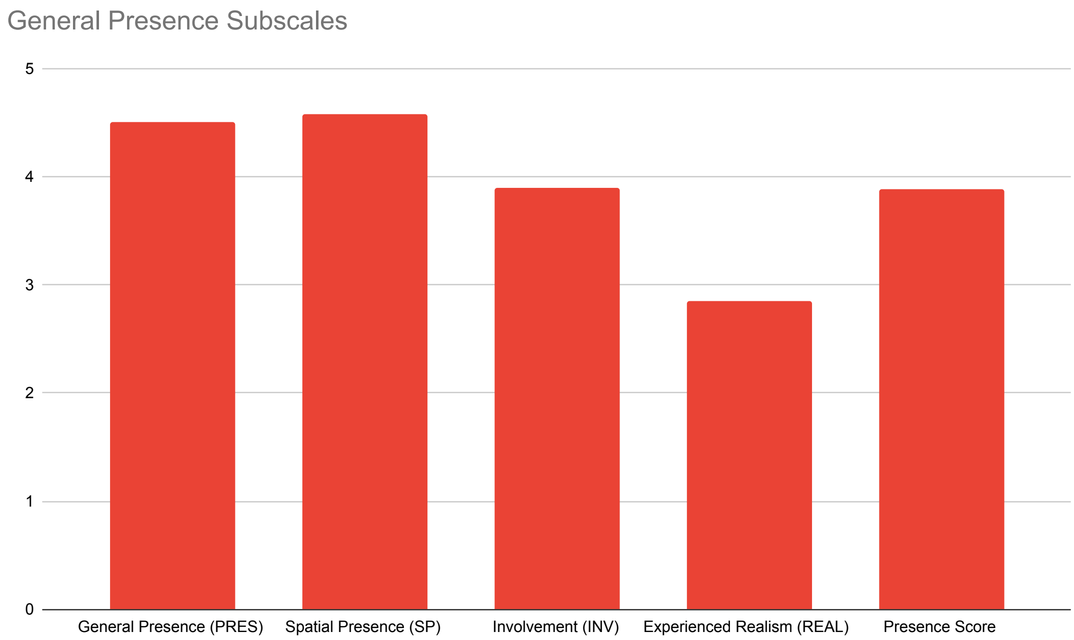


Figure 7.5: A bar chart over the general presence scores.

The presence score for the different tutorials in Table 7.2 shows that Tutorial C has the highest total presence score and an Experienced Realism score over 3, while Tutorial B has the lowest in all subscales as well as the total. Tutorial C was arguably the most diegetic tutorial out of the three, while Tutorial B had depth, sound and animations (aspects that we expected would increase presence) based on the analysis of presence factors in Section 3.1.2.

Presence scores tutorials	TutorialA	TutorialB	TutorialC
General Presence (PRES)	4.9	4.0	4.7
Spatial Presence (SP)	4.8	4.4	4.6
Involvement (INV)	3.7	3.4	4.6
Experienced Realism (REAL)	2.9	2.6	3.1
Presence Score	4.0	3.6	4.2

Table 7.2: Presence scores for each tutorial.

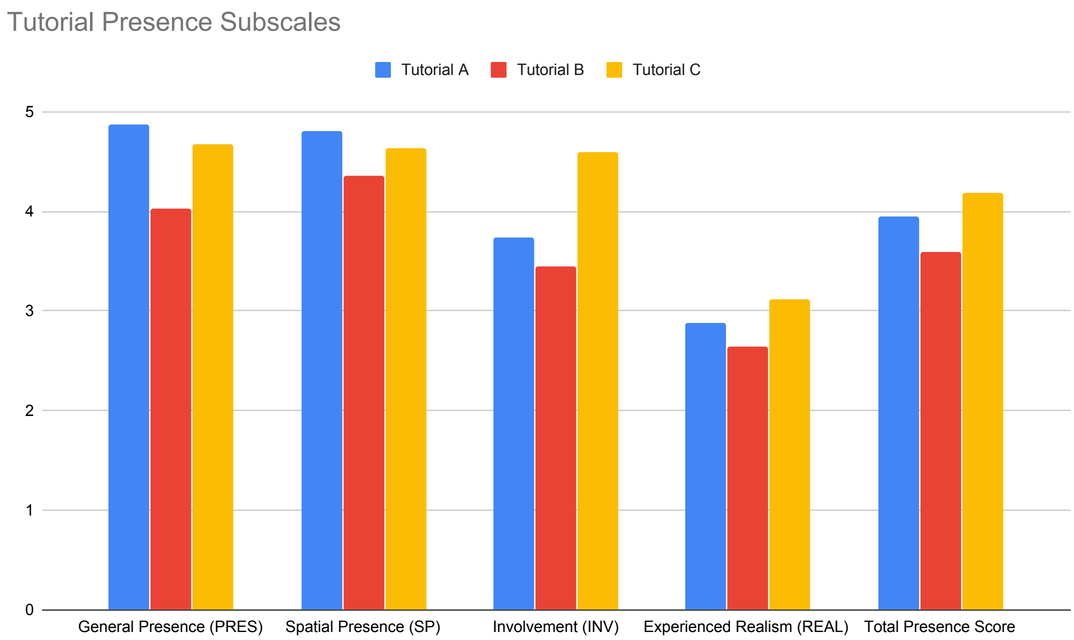


Figure 7.6: Presence scores for the tutorials.

The puzzles' presence scores in Table 7.3 shows that Puzzle 2 has the highest overall presence score and Puzzle 1 the lowest. This is interesting because it somewhat contradicts what the testers said in the interviews about the controls to Puzzle 2 being the most awkward and presence breaking among the three puzzles. As an example, one of the testers said this about Puzzle 2: *"So that broke [presence], both that it was an interface and especially, I think, that it was the wrong activity, or the wrong motion"*.[1]

[1]Translated from Swedish, actual quote: *"Så det bröt [presence] ju, både att det var interface och framförallt tror jag att det var fel aktivitet, eller fel rörelse"*.

Presence scores Puzzle	Puzzle 1	Puzzle 2	Puzzle 3
General Presence (PRES)	4.2	4.8	4.5
Spatial Presence (SP)	4.4	4.7	4.7
Involvement (INV)	3.7	4.3	3.8
Experienced Realism (REAL)	2.6	3.0	3.0
Total Presence Score	3.7	4.1	3.9

Table 7.3: Presence scores for each puzzle.

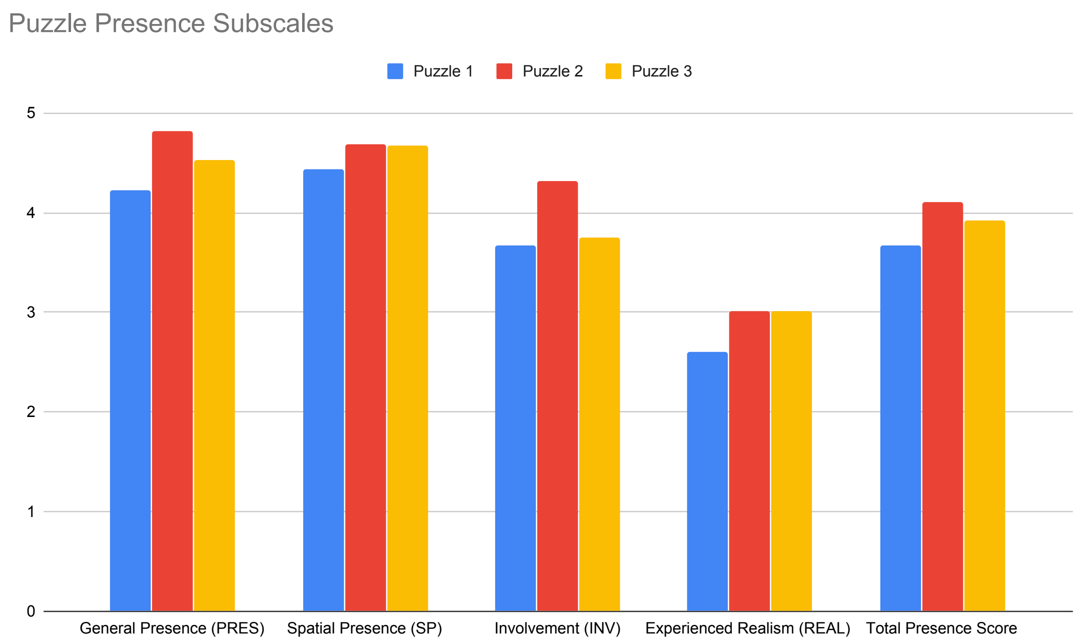


Figure 7.7: Presence scores for the puzzles.

In Table 7.4 we can see the presence scores divided by the order in which the users tested. The highest total presence score is reached during the second prototype they tested (this does not account for which tutorial or puzzle they tested). The first prototype they tested is the one that has the lowest score, while the second one had the highest. However, all of them are very close to the average and their differences are not statistically tested, so it's difficult to draw any real conclusions from this.

Presence scores Order	First	Second	Third
General Presence (PRES)	4.5	4.6	4.5
Spatial Presence (SP)	4.5	4.7	3.6
Involvement (INV)	3.6	4.2	4.0
Experienced Realism (REAL)	3.0	3.0	2.6
Total Presence Score	3.8	4.1	3.9

Table 7.4: Presence scores for in which order they played the tests.

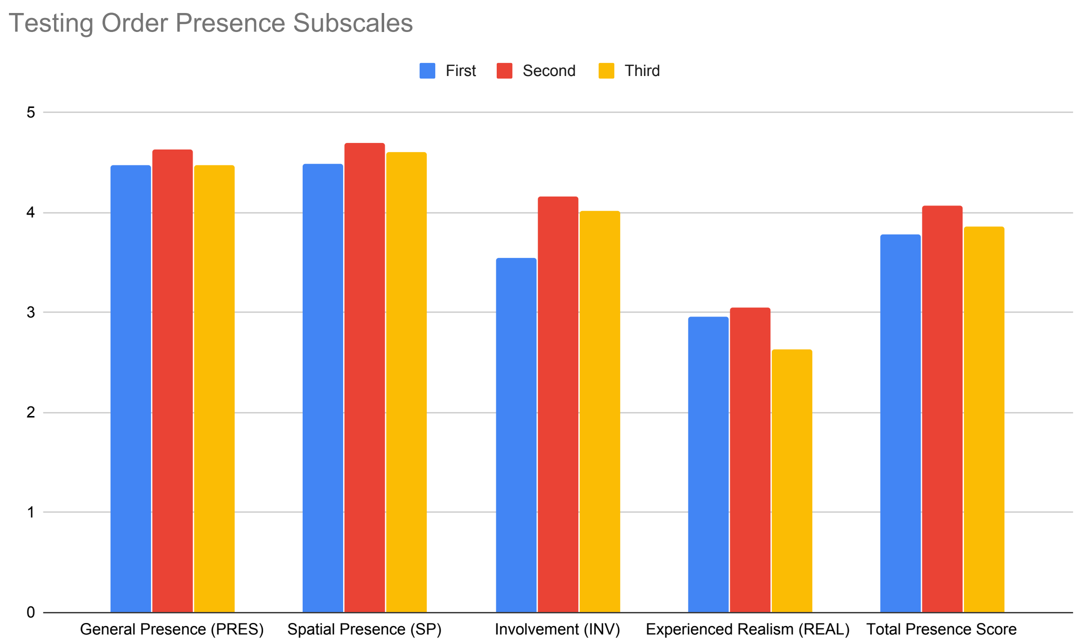


Figure 7.8: Order of testing presence scores.

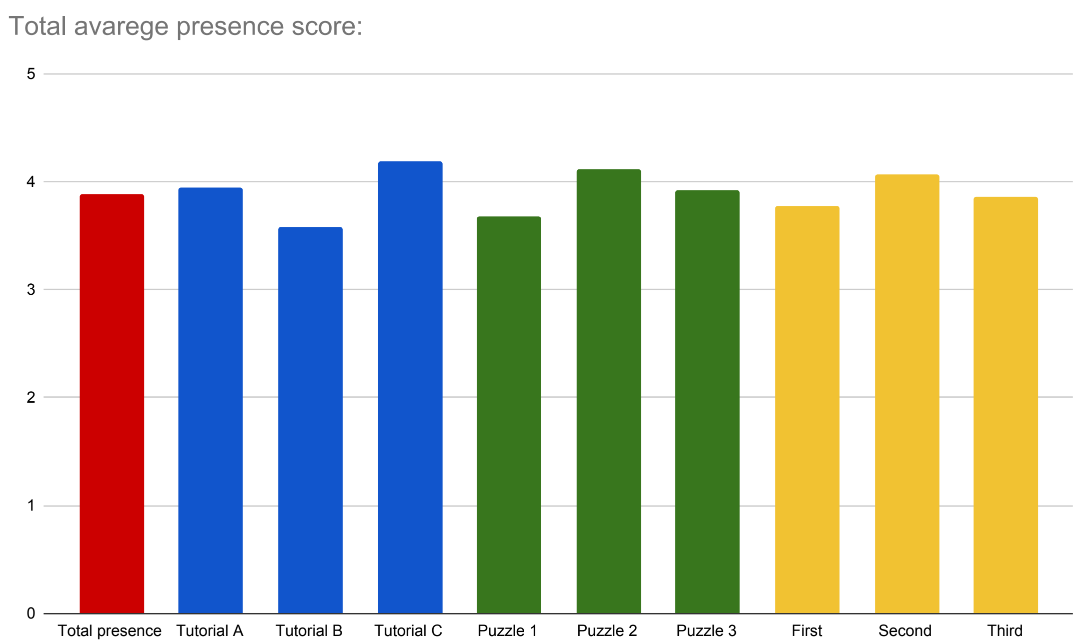


Figure 7.9: Average presence scores.

7.2.2 Data from coded interviews

When coding the interviews, we ended up with a lot of different codes and factors, connected to tutorials and puzzles. These numbers shows what codes that gave the most data, what the interviewees thought was the most interesting, and how large the data we analyzed was. However, since the main results from interviews are qualitative conclusions and guidelines (see Section 7.3), we will not go into details here, but more data is available in Appendix B, if needed.

After coding all of the transcribed interviews, we had a total of 267 coded quotes, with up to 3 codes per quote, ordered after priority. Each quote was also assigned a puzzle number and a tutorial letter, to better understand what each quote meant. Some of the quotes were ambiguous, as interviewees talked about several puzzles or tutorials at the same time, and was assigned a star "*" for partial ambiguity, or "All" when clearly referring to all at once.

In Figure 7.10, the total count of quotes that was coded for each code can be seen, divided among the three code priorities in each column.

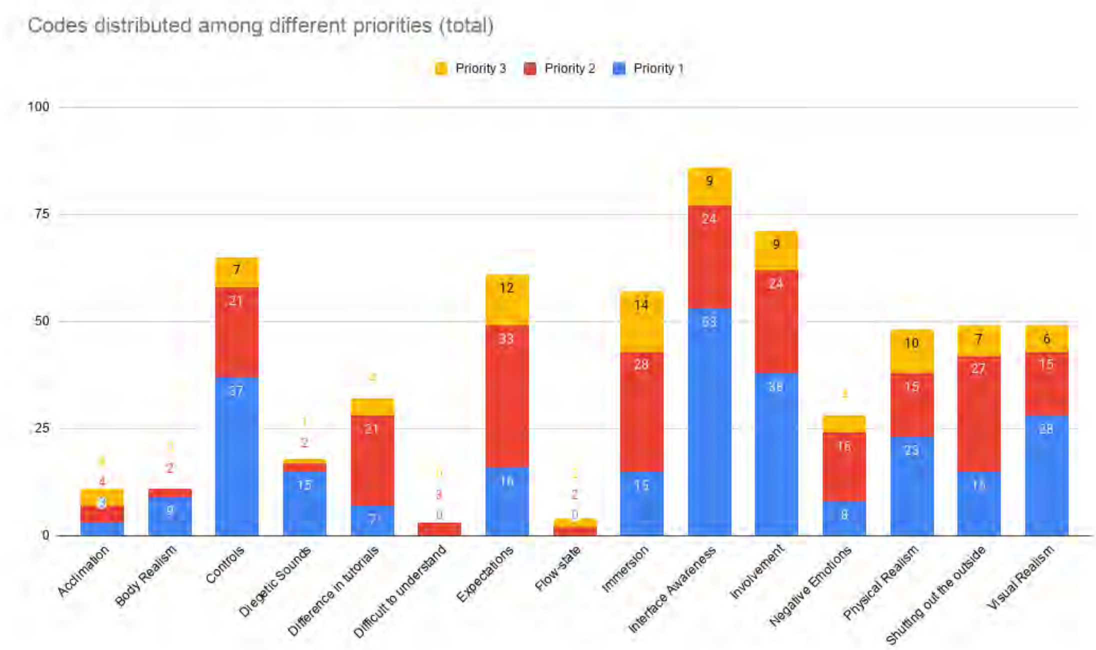


Figure 7.10: Chart over the total distribution of codes.

The following is a comparison of the quantitative diagrams of the coded quotes from the interviews, seen in Appendix B, with the qualitative results we got from analyzing the interviews. It's divided into tutorial-specific, puzzle-specific and general thoughts. The differences between the diagrams are often not very significant, though.

Tutorials

- Of all the quotes, Tutorial C is mentioned twice as much as Tutorial A. This could be because C is generally more interesting to look at, was more different than the other two, or sparked more thought. It could also be because Tutorial

A is much more boring or “normal”, i.e. similar to the style people are generally used to, when it comes to tutorials.

- “Interface Awareness” was mentioned many more times in connection to Tutorial A and C, than B, and C had the most mentions of all. We could interpret this as A and C being the extremes at two different ends of a scale, and that the unusual shape of C generated more thought. It could also be that, when we talked about different levels of diegesis, we might have mentioned A and C especially, and that C had the most interesting aspects for the testers to talk about.
- “Involvement” was mentioned 50% more often in connection to Tutorial B, than the others, which is interesting since a theory we had from the beginning, was that the more diegetic the tutorial, the more involvement or presence.
- Tutorial C was mentioned much more often in connection to “Difference in tutorials”, probably because if any of the tutorials, this would be the memorable one, on account of its different nature compared to the other two.
- When it comes to “Physical Realism”, Tutorial A was mentioned much less than B and C. This could have something to do with that B and C actually felt more real. On the other hand, one could argue that this would lead to A being mentioned *more*, rather than less, since it was less real.
- Tutorial A is mentioned very few times in connection to “Controls”, while Tutorial C is mentioned often. We have not been able to find a reasonable cause for this.
- “Acclimation”, or that the testers got accustomed to the environment, was especially strong in Tutorial B in comparison to the others. However, there were very few data points, and since this was an added emerging code during coding, we might have favored the others above this one and thus not used it as well as it could have been.

Puzzles

- “Involvement” was more evenly divided among the puzzles, than for the tutorials, which might say something about a lesser impact of each puzzle in comparison to the tutorials.
- “Controls” was mentioned often in connection to Puzzle 2, which seems logical, since that puzzle utilized a much more abstract control method than the others, which seem to have generated both frustration and afterthought in the testers.
- “Interface Awareness” was also mentioned often together with Puzzle 2, which might be because that puzzle had a non-diegetic interface, that the other puzzles did not have.
- “Physical Realism” was mostly mentioned in connection to Puzzle 1, which is reasonable, as that puzzle had a lot of physical objects that you interacted with

with your hands, so this might have enforced the feeling of physical realism better than in the other puzzles.

General

- "Visual Realism" was often mentioned when talking generally about all tutorials or puzzles, which is not strange, considering that the concept has a lot to do about the virtual environment or the general aesthetic, which was the same all through the different experiences.
- "Involvement" was also mentioned a lot for all tutorials and puzzles. This is probably because it's a very fuzzy and fluent concept, and there could be many different things affecting it in a virtual environment, while at the same time, it rarely has something to do with specific tutorials or puzzles.
- Mentions of "Shutting out the outside" is much higher generally for both all tutorials and all puzzles. This could be because, as for the concept immersion (not the code), it is the environment and outer factors that affects this, rather than specific elements in the virtual experience.
- "Diegetic Sounds" is often mentioned in connection to the whole experience – despite sound effects being connected to specific objects or mechanics. We believe this could be because all the puzzles had the same sort of sound effects applied to objects. That Tutorial C was mentioned more often, however, could then be that its sound effect actually was diegetic and stood out from the others.
- Generally, all puzzles or tutorials seem to be mentioned more often than each one specifically. An exception to this is "Negative Emotions", where it was more evenly distributed. This might be because negative emotions many times comes from very specific reasons, which we then might have been able to easier connect to certain puzzles or tutorials when coding.

7.3 Qualitative analysis

The main purpose of this study was to evaluate which factors to consider, and how, when designing tutorials for VR. First, we analyzed and evaluated the quantitative results in the previous sections, and learned a lot about the users' presence and reactions to different aspects in the prototypes, and compiled this in a qualitative factor analysis. We then boiled these conclusions down into specific guidelines that are related to tutorial design for presence in VR, specifically, to answer the research question.

7.3.1 Factor analysis

These are the conclusions from our qualitative analysis of the coded interview data, summarized for each presence factor (see Section 3.1.2 for definitions of those).

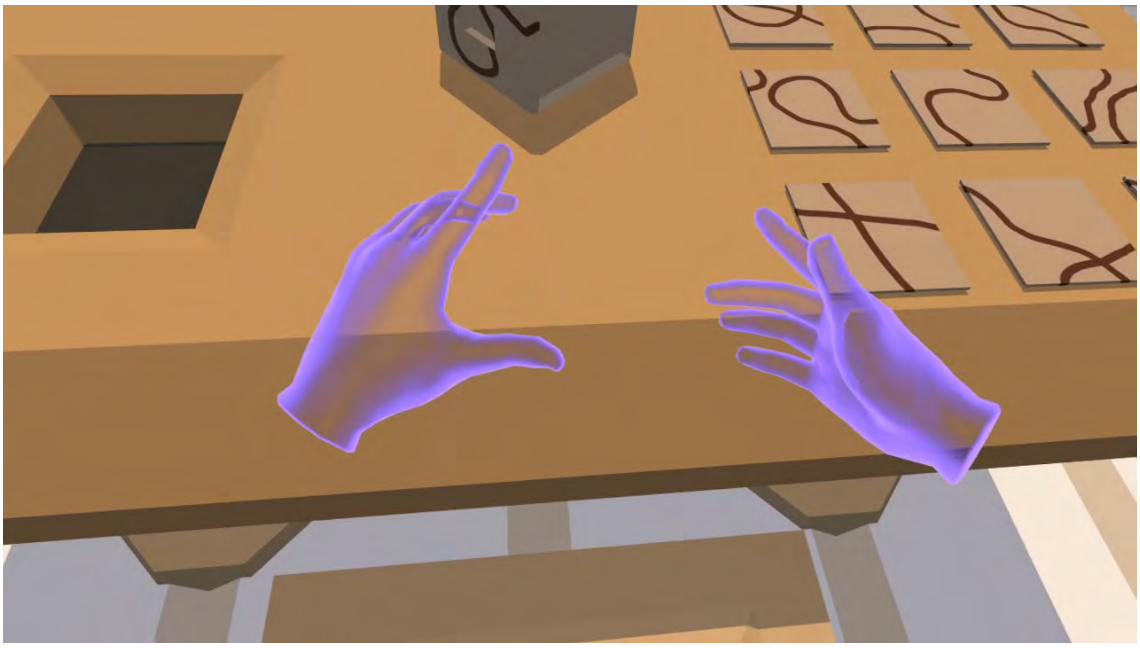


Figure 7.11: The hands of the player.

Note that these conclusions cover most presence-related aspects that we found, and not only those relevant to interfaces and tutorials. For a summary of the design guidelines for working with presence and tutorials in a virtual environment, see Section 7.3.2.

We had 16 users test the prototypes and the following analysis is based on statements from all or some of these.

Body realism

The tested experience was mainly body-less, except for the hands of the player (see Figure 7.11), so this is generally what people commented on. One said: *"It's one of these things that, when it works, you don't think about it, of course my fingers are suppose to be like that; but when it doesn't work, it's like, ew, now it feels like I'm holding . . . I'm operating something. . . "*.[2] According to a about half of the testers, when the hands looks real or behave in a realistic manner when grabbing or moving the fingers, and so on, you don't think about it. Consequently, when something feels off about the hands or their movement, this break presence. Other studies on the subject will agree on that a good virtual body representation increase presence[24], and that unrealistic bodies or body movement decrease it[32].

A couple of testers did also comment on their inability to for example stand up from the chair and move about the room, which made them feel limited or even trapped. We are not sure if this affects presence, though, since it might just have been a desire to explore, rather than a direct need, which would mean that it's not just a

[2]Translation from Swedish, actual quote: *"Det är ju en sån där grej som, när det funkar så tänker man inte på det, det är ju klart att mina fingrar ska ligga sådär. Men när det inte funkar så är det ... Ew, nu kändes det som att jag håller i . . . I'm operating something. . . "*

negative aspect of the environment. Riches et al. do however bring up a negative effect on presence when feeling restricted in some way[32].

Controls

Basically, everything about the controls that might be a cause of irritation or frustration could be things that break presence. According to almost all testers, this happens mainly when, as a user, you think that you are giving the correct input or doing the correct movement for a certain task, but you don't get the expected result. A majority also reported that it happens when the controls or interactions are not intuitive or hard to understand, which can cause the experience to feel weird or unnatural (as some commented on Puzzle 2, see Figure 6.10). A third of the testers said a cause of this could be that the control methods are so complex or advanced, that you have to think hard before using them in order to do something correctly, which might make the abstraction level between you as a user and the virtual world much more apparent. A third of the testers also reported this when the controls are limited or doesn't allow you to do everything you expect, sparking frustration or irritation, which breaks presence. One tester phrased it like this: *"Yeah, I thought the controls for that cube that you were supposed to switch layers on, was mostly annoying. It felt like it was working against me"*.[3]

On the other hand, when control methods are intuitive and a user is able to interact with the virtual environment without thinking about it, their presence will increase, according to almost half of the testers. The same goes for having concrete and physical interactions, more than half of the testers said, such as picking up or moving objects (see Figure 6.9), or pulling actual levers or pushing actual buttons, and so on. This gives a natural, intuitive feeling of a real and working world around you, enforcing presence. These notions are supported by several studies that ensures the importance of both degree and mode of control[15, 23], as well as body engagement[24, 33], when it comes to presence.

Even though not thinking of the mechanics of a VR experience is almost always a desired outcome, sometimes explanations are needed to let a user get more familiar with advanced concepts, according to a third of the testers. In these cases, a couple of testers reported that, although there might be a break in presence early on, the more a user can get comfortable and acclimatized to the more advanced concepts, the more they feel present in the world again.

Diegetic sounds

Sound is clearly important for presence, according to our findings. Ambient soundscapes create an atmosphere that partly shuts out the real world, and partly helps to create a mood and feeling for the virtual world. The same is valid for sound effects for specific elements in the environment.

The most important part of audio in VR, says more than half of our testers, is

[3]Translated from Swedish, actual quote: *"Ja, jag tyckte kontrollerna på den här kuben man skulle byta lager på var mest irriterande. Det känns som den arbetade emot mig"*.

that it doesn't stand out from the rest of the experience in a bad way, or create unnecessary friction when not wanted. Mostly, users don't think about the sound when it feels like a part of the environment, but as soon as something didn't sound as expected, several testers reacted considerably. This include both when something didn't have a sound at all, but was expected to; and when something did have a sound, but didn't fit into the context. This follows much the same conclusions as other studies before us[24, 15, 33].

Regarding diegesis specifically, it may not be as important as we thought at the start of the project. As long as sounds, both ambient and sound effects, don't stand out too much and behave as expected, non-diegetic sounds does not seem to be a problem. A tester said this about the sound design: *"The sound got me more present, because then I couldn't hear the real world, but it was so calm and soothing, so I didn't notice it more than that it just was there, and that got me more present".*[4]

Expectations

We bring a lot of expectations with us into a virtual world, but just as many are created by everything we experience in it. Among both of these kinds, there are many things that can be anticipated and controlled.

A general rule, supported by all of the testers, is that when something is not behaving as a user expects it to, there's a chance that it will break presence. One tester said: *"Had it been like that, but I would have grabbed it and turned: that might have worked as I had thought, then I probably wouldn't have been bothered by it".*[5] This could be when the user expects something to happen as a reaction to their own behavior, but nothing happens, or on the contrary, that the user does not expect anything to happen, but they are surprised by something that is caused without their knowing. It could also be that a user perform an action and expects a certain kind of results, but instead get something entirely else. In all these cases, the user might be jerked out of the experience, and the cause for these breaks of presence is often frustration, irritation or confusion (see factor Negative Emotions for more details). Indeed, Witmer and Singer also underline the importance of matching user anticipations when working with presence[15].

Of course, this is not always the case. Sometimes, surprises can be something positive, or at least just part of the game and nothing that users react strongly upon. An example of this was reported by a third of the testers about the surprising rain of confetti at the end of the experience (see Figure 7.12). However, it's still important to be aware of the potential consequences of breaking expectations, just the same.

[4]Translated from Swedish, actual quote: *"Ljudet fick mig att vara mer närvarande, för att då hörde jag inte den verkliga världen, men det var så pass lugnt och rogivande, så att jag la inte märke till det mer än att det fanns där och fick mig att vara mer närvarande".*

[5]Translated from Swedish, actual quote: *"Hade det varit den grejen, fast jag hade grabbat tag i den och vridit: det hade nog funkat som jag hade tänkt, då hade jag nog inte stört mig på den".*

Figure 7.12: The confetti rain after finishing a puzzle with Tutorial C.

The opposite of defying expectations could arguably be to create a safe environment and build up self-confidence in the user. In fact, a third of the testers experienced something similar, and it seem to have strengthened their presence along the way. Riches et al. also bring up the positive effects of familiar environments in a recent study[32]. It should be pointed out, though, that just to avoid defying expectations is not the same as creating a safe environment, as the former usually only lead to users not losing presence, and nothing more, based on statements from almost all of the testers, for example: *"In the first and second test, I didn't need to think about it, everything just worked as I imagined it would"*.[6]

According to some of the testers, part of working with expectations can involve making sure that instructions in VR are clear and simple, that the virtual world is consistent and follows its own rules, and that new concepts are introduced in familiar surroundings.

Immersion

Since we did not control this aspect during the experiments, as each test was performed in the tester's own home at their own conditions, the resulting comments about it was somewhat sprawling and inconsistent. None the less, this is what we found out about the effect of immersion on presence.

First and foremost, when a user interacts with the outer, real world, this breaks presence, according to half of our testers, who experienced this. Among those, this tester said: *"I think I hit the table at some time. You get kinda surprised when*

[6]Translated from Swedish, actual quote: *"I första och andra testet så behövde jag aldrig fundera, där funkade allting som jag föreställde mig att det skulle göra"*.

you're there, 'cause you don't see anything else".[7] It could be by mistake, such as accidentally hitting the desk or computer screen in front of them, or it could be deliberate, such as needing to take off the VR headset to adjust in-game sound volume, or wipe the lenses. In either case, this makes the user more aware of the real world, and thus weakens their presence in the virtual one, which Slater and Steed also confirms in their study[30]. The former is of course hard for a developer to avoid, other than reminding the player to keep the playing area clear, but the latter could possibly be avoided by extensive testing, so that in-game volume is at a good level, text is easily readable, and other aspects that would need to be fixed or checked outside of the experience.

A specific case of this is the controls used to interact with the virtual experience. Making sure that a product works smooth and friction-less with all supported controllers is clearly important. Additionally, the way controls specifically works in VR is just as important, as unnatural or cumbersome interaction that work in ways the user don't expect, will likely break their presence as well, half of the testers reported. This could perhaps be avoided if a tutorial taught these controls simple and smoothly, as well.

Other than that, the interactions in VR can be a source of breaks in presence. It could be that a user faces a puzzle or problem that they feel is close to impossible to solve or get around, according to a third of the testers, which then might give them a feeling of "playing a regular game", which of course is not exactly the same thing as being present in a virtual environment.

Interface awareness

In the context of presence in VR, uncertainty is rarely positive. If a user is unsure if something is working or just how it's working, they tend to get more aware of their controls and interfaces in the environment, according to two thirds of the testers. *"Oh, this is a 2D-UI, something that's digital, that's not, like, real"*[8], a tester claimed to have been thinking during a test. Half of them told us that, when an interface stands out or doesn't fit into the rest of the world, it also makes the user very aware of that it actually is an interface and not an integral part of the virtual world, which might break presence. Conversely, if either interface or interaction makes sense and fit into a certain VR environment, the user doesn't give it much thought. This correlate a lot with what Witmer and Singer say about interface awareness and the consistency of multimodal information[15].

The content of interfaces also matter for presence. Two thirds of the testers report that, when instructions are unclear or difficult to follow, or just hard to read, users react negatively to it, which breaks presence. Having to read longer texts seem to be generally boring, according to half of the testers, which can be a cause of frustration and breaks of presence. This is further backup by Sutcliffe, who argues that

[7]Translated from Swedish, actual quote: *"Jag tror jag slog i bordet någon gång. Man blir typ förvånad av när man är där, man ser ju inget annat".*

[8]Translated from Swedish, actual quote: *"Jaha, det här är 2D-UI, nånting som är digitalt, nåt som liksom inte är verkligt".*

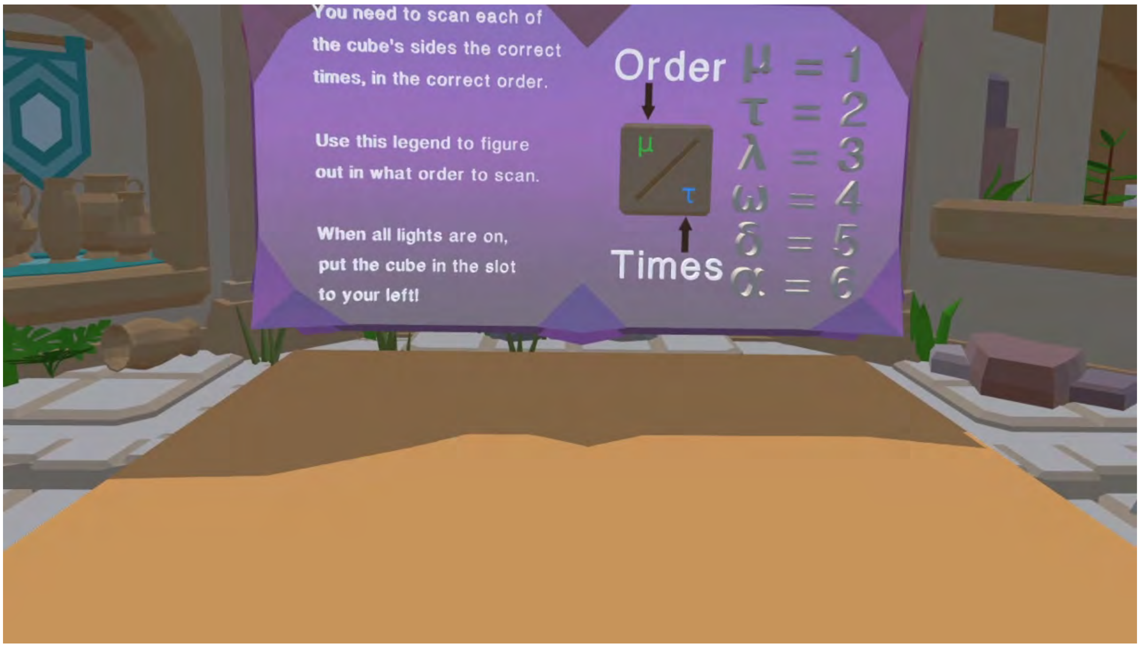


Figure 7.13: The instructions to solve Puzzle 3

uninteresting or mundane content bores readers[75]. Short and simple instructions that does not need to be shown more than once or twice, if even at all, appear to be something to strive for.

Another thing that became apparent during testing, is that users focus much more on the contents of an interface, such as instructions or other information about the experience, than the actual look of the interface, according to more than half of our testers. One of them said: *"In the last one I didn't focus what it was because I was much more focused on the text, because it was more text"*. In fact, when not focusing on the interface contents, users rather focus on what they are doing themselves, instead of the interface, additional testers testified. This suggest that the look-and-feel of an interface, although important for the general mood of an environment, does not seem to have that much impact when it comes presence, in comparison to how the interface is used or what it contains. This is also supported by Nunez, who argues that the content of non-diegetic music have significant effect on a user's sense of presence[35].

The above became very clear when asking each tester if they noticed any differences between the visually very distinct three tutorials (see Figure 7.14), and the majority of them never even noticed a difference, and those who did, didn't pay it any attention.

Involvement

The strongest conclusion that emerged about this factor, was that the more focused a user is on what they are doing (a puzzle in our case), the less they payed attention to their surroundings, including interfaces or tutorials, based on statements from almost all of the testers. One of these statements went: *"I noticed that, every*

Figure 7.14: To the left is Tutorial B and to the right is Tutorials C, both with the same content but different looks.

time I got something to focus on and distract myself with, I got less aware of the real world".[9] This is generally a positive state, as the user gets so engrossed in what they are doing that the surroundings lose importance, and could perhaps have something to do with flow-state, since this effect is just as real outside of VR. In a study by Johnson and Wiles, correlations between flow-state and enjoyment of games are considered in much the same manner[25]. Slater and Steed also argues that the opposite; loss of attention, likely will cause breaks in presence[30], which fits our results as well.

What exactly triggers these states of focus, however, is less clear. It seems like it could be anything as long as the user finds it interesting and engaging. Examples of such triggers include (with a couple of reports per each one): having tactile puzzles where grabbing and moving objects with their own hands is part of the problem solving process; having a high enough challenge that triggered users to overcome them; when the world is full of interesting elements that draws you in; to learn about new things (both mechanics and concepts, as well as story-wise) by doing or experimenting, and not only by reading about it; to name a few. This is somewhat tangential with Witmer and Singer's claims that the more meaningful an experience is perceived by the user, the more presence they feel[15]. Additionally, the impact of an engaging narrative is also mentioned in a study by Riches et al.[32].

A backside of being totally engrossed in a certain activity could be that other interesting or engaging details in other places can be missed, based on different assertions from testers. This might both mean instructions or important details, or things that may not be crucial for the progression but on the other hand makes the environment more alive and thus enhancing the user's feeling of presence.

This also connects to the fact that photo-realism is rarely needed when building up presence, but rather interesting elements in a visually appealing and welcoming environment (as perhaps seen in Figure 7.15). See factor Visual Realism for more details on that.

[9]Translated from Swedish, actual quote: *"Jag märkte att, varje gång jag fick någonting att fokusera på och distrahera mig själv med, så blev jag mindre medveten om den verkliga världen".*

Figure 7.15: Part of the environment in the prototype.

Negative emotions

Mainly, we found that strong negative emotions such as frustration, irritation, confusion or uncertainty could impact presence negatively, according to half our testers, for example this one: *"I guess it was when I needed to redo the scanner puzzle several times. This made me irritated, and that broke presence".*[10] These emotions are all vastly associated with a lot of things under the other factors that supposedly could break presence. Although they do not mention the exact same type of feelings, Slater and Steed also support the idea that negative emotions often might break presence[30].

However, negative emotions does not always seem to be bad for presence. A couple of testers testified to have been feeling performance anxiety or a pressure to perform as fast as possible (even though we were very clear on that they were welcome to take as long as they needed). One of them said: *"Also later, in the second puzzle with the lights and the scanning, I felt, I don't know if it was stress but it was some kind of pressure, almost performance anxiety. I thought, shit, I got four lights, I can't ruin this now, my killstreak. And that's actually things that made it more real for me".*[11] This apparently enhanced their presence, letting the puzzle take their focus completely. Even more interestingly, one of the testers even felt greater presence while getting frustrated about certain parts of the puzzles, because it didn't work as they expected it to, but they instead felt themselves becoming obsessed with finding

[10]Translated from Swedish, actual quote: *"Det var väl när jag behövde göra om scannerpusslet flera gånger. Då blev jag irriterad, och då bröt det presence".*

[11]Translated from Swedish, actual quote: *"Sen också i andra pusslet med de där lamporna och scanningen där kände jag, vet inte om det var en stress men det var lite som en press, en prestationsångest nästan. Att jag, shit nu har jag fyra lampor, nu får jag inte förstöra det här nu, min killstreak. Och det är ändå saker som gjorde det mer verkligt på något sätt för mig".*

a solution instead. This last one is of course not a typical case, but still interesting to mention. A recent study by Riches et al. also showed that for example social anxiety, paranoia or loneliness can increase presence, as well[32], further backing up this idea that negative emotions not only breaks presence.

Tangential to involvement (see factor Involvement), story-based experiences often work with both positive and negative feelings to create an interesting and involving narrative, because a story without the dark parts is often experienced as utterly boring.

In summary, it's not as easy as just avoiding everything that might spark negative emotions, when creating a virtual environment. Elements of different kinds would have to be considered and tested extensively, to be sure that the sought after reaction and experience was reached among users.

Physical realism

When things in a virtual environment work as people expect them to, or just in a realistic manner, it either enforces a feeling of a realistic environment, or they don't think about it at all, and in either case, it amplifies their feeling of presence, according to two thirds of the testers. Among those, one said: *"But otherwise it was just as immersive, I think, because it still worked as I expected it to work in reality"*.[12] And conversely, almost as many testers ensured us, when things does not work as expected or in an unrealistic way in a virtual environment, people react and starts to think about the inconsistency, breaking presence on the way. Other studies also supports this notion about the importance of physical realism[33, 30, 32, 23].

Based on a couple of more specific statements, this doesn't necessarily mean that everything needs to be realistic, but rather that the rules of the virtual world needs to be clear and consistent, so that a user can feel safe in knowing how things work. Scepticism towards the virtual world around you tend to break presence (see factor Expectations for more on that). So magical or otherworldly solutions might work well, as long as it fits the context. For the same reason, it's important not to unnecessarily create frustration in the user, since this might raise their guard towards the rules of the virtual environment.

The very concrete and physical interaction of picking up objects from the table and combining them, and so on (see Figure 6.9), enhance users' sense of presence, according to a third of the testers. The opposite is also true, even more testers told us: when users for example can't reach something, or the interaction is unclear or awkward, they experienced breaks in presence. This also supported by Witmer and Singer, who argue that presence will increase, the more physical environmental modifiability a user experiences[15], as well as Slater and Wilbur, who underlines the importance of bodily engagement[33].

[12]Translated from Swedish, actual quote: *"Men annars var ju det lika immersive, tyckte jag, för där funkade det ändå så som jag förväntade mig att det skulle funka i verkligheten. På precis samma sätt som den första"*.

Shutting out the outside

Generally, it seems like having something to focus on in the virtual environment, makes it easier to shut out the outside and outer stimuli, almost all of the testers told us. For example focusing on a challenge, according to one tester: *"When the puzzle is hard, I don't really pay attention to anything else, so it doesn't really matter either way"*. Having a well designed and embracing sound design (both ambient and sound effects) would also help, says some (and are backed up by Slater and Wilbur[33]), or if the surrounding environment feels pleasant and inviting, according to several testers. Witmer and Singer talks about selective attention being a positive presence factor, which is basically the same thing[15]. Slater and Steed, on the other hand, argues that loss of attention causes breaks in presence[30], which supports this as well.

A related aspect is that of feeling safe in an environment. When the virtual experience is inconsistent or just hostile, a couple of testers says, the user is less inclined to open up to it, and tend to be more aware of the outside world. Similarly, when feeling frustration over mechanics or other difficult aspects of the construct, others say, it's easier to start thinking about how things work from an outside perspective. Things like these easily break presence.

Visual realism

It seems like visual realism, for example the environment feeling realistic and like a real place, would enforce presence, according to half of the testers. However, it is important to note that "realism" does not mean the same thing as "photo-realism" (the difference can be seen exemplified in Figure 7.16). It rather means that the virtual world is consistent and that things behave as one expects them to, which several testers point out, although photo-realism may of course help as well. One of our testers told us: *"That [Keep Talking and Nobody Explodes] doesn't have better quality than you have, and in that I feel extremely present. Same thing with Job Simulator. I don't think you need photo-realism or similar, to achieve hundred percent presence, either"*.[13]

This is further supported by several studies, that either examined scene realism[15], or failed to find any difference in presence when it comes to different levels of photo-realism[24].

As long as visual details and the general environment feels possible and that they fit together, users tend to be satisfied. On the opposite side, when something visual don't fit the general context, users express disbelief of the virtual world. So when something doesn't feel realistic, users seem to have felt less presence, says a third of the testers, for example like this: *"Also for the scanning part, which was the third one, I would have prefered a different environment. I would have prefered if I was*

[13]Translated from Swedish, actual quote: *"Att [Keep Talking and Nobody Explodes] har ju inte bättre kvalitet än vad ni har, och där känner jag ju en extrem presence. Samma med Job Simulator. Jag tror inte på att man behöver ha fotorealism eller sånt, för att man ska uppnå hundra procent presence heller"*.

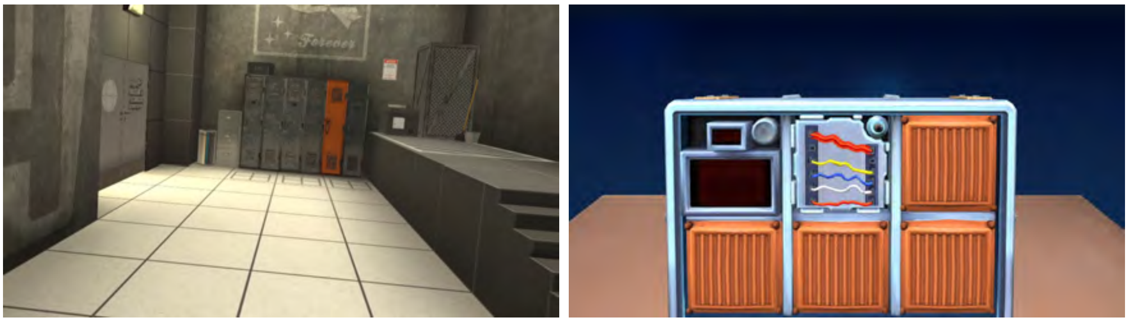


Figure 7.16: To the left the more photo realistic *Boneworks* and to the right more stylized *Keep Talking and Nobody Explodes.*

in like a supermarket or so, because then it would have made more sense what I'm doing with respect to my surrounding"(see Figure 6.11).

Another interesting aspect that surfaced, was that when a couple of testers took initiative to look around the environment and explore it more than required by the rather straightforward puzzles, they felt more presence. Thus, they told us that they wished that the experience had encouraged them to look around more. This is something that Witmer and Singer also mentions about active search[15].

On a similar note, several testers seem to have enjoyed the rain of confetti at the end of each puzzle (see Figure 7.12), even though it was a rather unrealistic feature. Instead of unease or disbelief, many expressed joy of seeing the confetti rain down towards them, and some of them even raised their hands to meet it. Some other testers, however, reacted more on the lack of realism of the confetti, so features like this might be complicated to work with.

When objects in VR appear in similar fashion and position, such as the chair the user sits on, or the table in front of it, people seem to have a positive response to this matching, a third of the testers reported. Although, some testers al expressed a break in presence when they discovered differences between them, such as spinning around on their own swivel chair not matching the solid and not spinning chair in the virtual environment.

7.3.2 Tutorial design guidelines for presence in VR

In Section 7.3.1, our qualitative findings are laid out for each of the derived presence factors (described in detail in Section 3.1.2). Here we give more of a summary of that, entirely focused on the design of tutorials and interfaces, rather than the factors themselves, since this is the main focus of this thesis. Note that this is a considered selection from the qualitative analysis and therefore doesn't cover all of the factors extensively (see Section 7.3.1 for that).

The main body of this section is the following list of all noteworthy conclusions as 20 guidelines for designing tutorials (and interfaces) for virtual environments. These are mainly based upon the resulting data of the factor analysis, but also leans on a general understanding of the field and the derived presence factors that we read up on during the pre-study. For each guideline, the basis factors and literature of our conclusions are stated, to make it clear how each guideline was derived. The guidelines are also categorized after how sure we are of them, starting with the strongest.

At the end of this section, we'll also sum up what to focus on or prioritize, among these.

7.3.2.1 Very strong guidelines

These 4 guidelines are based on conclusions that all or almost all of the testers agreed upon.

A tutorial should function in the way a user expects it to

A tutorial should function in the way a user expects it to, and not advertise otherwise. Note that the nature of a tutorial in this case will matter, for example a diegetic tutorial might advertise a completely different interaction than a non-diegetic one.

This has some ties to the factor Interface awareness, but is mainly based on the findings about factor Expectations in the factor analysis. Witmer and Singer also support this[15].

Aim for intuitive controls and interactions

When a user can intuitively interact with the virtual environment without thinking about the controls, it increases presence. When the controls are not intuitive or work in an unexpected manner, it may feel weird and unnatural to the user, which can break presence, instead.

This can be clearly understood from the analysis of factor Controls in the factor analysis.

Design for natural and realistic interfaces

When things work in a realistic way, or as the user expect them to, they generally do not think about it. If it does not work as expected, the user will react to it, which possibly breaks presence. Additionally, going against expectations can cause confusion, frustration and irritation, which also might break presence.

One should therefore design interfaces and interactions to be natural and intuitive. This implies making tutorials as clear and easy to follow as possible. Furthermore, since the controls of a virtual experience is an abstract layer between the user and the virtual environment that just needs to work as expected; basically, the best tutorial is one that is not even needed, when everything just works as expected instead.

Our findings about both factors Expectations and Physical realism in the factor analysis strongly support this. Several studies has also come to similar conclusions[33, 30, 32, 23, 15].

A strong focus makes the user forget about their surroundings

Focusing hard on something specific in a virtual world, often makes the user forget about their surroundings, both real and virtual. This can be used for benefit, but it may also get in the way of making sure that a user learns something specific, because they may focus too hard on something and not pay attention to a tutorial or the information it contains.

Almost all of the testers mentioned this in some way (see factors Involvement and Shutting out the outside in the factor analysis). We also found that a couple of studies supported this conclusion[30, 15].

7.3.2.2 Strong guidelines

These 4 guidelines are based on conclusions that about two thirds of the testers agreed upon, or about half of them but with a strong basis in literature, as well.

Interfaces that does not fit in makes a user more aware of them

Interfaces that functionally or visually do not fit into the virtual world, makes a user more aware of them, which might break presence. This should therefore be avoided.

See factor Interface awareness in the factor analysis for details on our basis for this, and it's also backed up by Witmer and Singer[15].

Make all information as clear as possible to avoid confusion

When instructions are complicated or hard to follow, users either become more aware of the interface, or just irritated and frustrated, which breaks presence in either case. Simple and clear instructions, on the other hand, help users to understand when something goes wrong and will in the best of cases not be needed more than once. Generally, one should strive to make all information presented to users as clear and articulate as possible, to avoid misunderstandings and confusion further on.

This was clear from the factor analysis (see factor Interface awareness), and parts of the reasoning is also grounded in a study by Sutcliffe[75].

Favor concrete controls over complicated or abstract ones

Concrete and tangible interaction in VR, for example using their hands to move objects around or pull levers, lets a user feel like interacting with the virtual world directly, which strengthens presence.

Abstract or non-intuitive controls, when the user has to try to remember what the controls are, or when the user starts to think about how the controls really work; this makes it clear for them that they are not interacting with the world directly, but instead through an abstraction layer of hardware, and this line of thought can break presence.

This was quite clear from the factor analysis (see primarily factor Controls, but also Immersion), and there are also several studies that support this in different ways[15, 23, 24, 33].

A solid and coherent sound design is important

Sound can play a big role when it comes to shutting out the real world, both ambience and sound effects. It's also important that certain things really sound like the user expects them to, otherwise it might break their presence. At the same time, it's important not to add sound effects or ambience that stand out too much in contrast to the rest of the experience, or sound like they do not fit, since getting startled by this will break a user's presence as well.

Clearly, this comes from analyzing the factor Diegetic sounds, but it's also tied to Shutting out the outside. There are several studies supporting this as well[24, 15, 33].

7.3.2.3 Less strong guidelines

These 4 guidelines are based on conclusions that about half of the testers agreed upon.

Tutorial contents are more important than visual presentation

The instructions and contents of a tutorial is more important than how it is visually presented, since most of the users' focus tend to be at the interface contents or what they themselves are doing, instead of concentrating on the visuals.

This comes from the analysis of the factor Interface awareness. Nunez also have some interesting findings that might correlate to this[35], as mentioned in the factor analysis.

Avoid long texts or too much information at once

It's frustrating to be forced to read a lot of text, which in turn can break presence. Being overwhelmed with too much information or impressions at once can also break presence. This suggests that one should be conservative with text and information in tutorials, and perhaps rather put energy in trying to convey the intended message in a way that doesn't overwhelm or bore the user.

This is first and foremost based on our findings about the factor Interface awareness, but also ties in with Negative emotions. Sutcliffe also clearly support this[75].

A coherent virtual world can be more important than photo-realism

Making sure that the different visual elements all fit together in a coherent virtual world, and that this world also match the activities that the user is engaged in, seem to be much more important for making a virtual environment feel realistic, more than focusing on the level of photo-realism in the world.

See factor Visual realism in the factor analysis for more details on that. A couple of studies also supported this notion[15, 24].

Use interface sounds with care

Having sounds in an interface gives the user good feedback, but they should fit the virtual world and experience. If they do not, users will start to recognize what the interface is, and think about the mechanics behind it, which will take them out of the experience. However, having *diegetic* sound effects, as opposed to non-diegetic ones, does not seem to be as important as it is to make a coherent experience, regardless of the level of diegesis.

Much like the guideline “A solid and coherent sound design is important”, this is largely based on the analysis of the factor Diegetic sounds, but also connects to Interface awareness.

7.3.2.4 Weaker guidelines

These 8 guidelines are based on conclusions that less than a third of the testers mentioned. For some, there might just be a couple of testers agreeing on it, but all these guidelines are considered interesting and helpful enough to be included, even so.

Avoid creating scepticism or uncertainty about the environment

When the user feel insecure or unsure, for example when something unexpected happens, it creates scepticism and uncertainty about the environment, which can break presence. Conversely, feeling confident and safe will give a sense of continuity and consistency.

It then follows that it’s important not to surprise the user too much (if not intentional, of course). Recognizing a place or being used to something, are some of the things that might create confidence and a feeling of safety in a user. A way of using this for tutorials, is to introduce new concepts in a safe environment that the user is already used to, to not break presence.

This is mainly based on our findings about factor Expectations in the factor analysis, but is also related to the factor Negative emotions. A recent study by Riches et al. can back this up as well[32].

Get users acclimatized with problematic aspects as soon as possible

When controls or tutorials are awkward or unnatural in the beginning, it does get better the more the user is acclimatized to them, so the faster they get used to it, the better. Similarly, when interactions or the virtual world does not work as the user expects them to, it’s important to have relevant tutorials or instructions as early as possible, to make it easier for a user to familiarize themselves with an environment.

This is based on statements from a couple of testers, as described in the analysis of the factor Controls, but it also ties in with the factor Expectations.

Let users figure out some things by themselves when possible

One way to achieve simple or minimalist tutorials, is to completely or partially let the user experiment by themselves as a way to understand the issue at hand. Indeed, not giving the user all of the information at once, but letting them figure some things out by themselves, can increase involvement in a task – as long as the learning curve is not too steep.

However, to experiment for understanding is not always viable, since some things are not possible to guess. In these cases, it’s important to keep it simple, intuitive and consistent. The shorter, less information-heavy tutorials, the better.

This conclusion comes from the analysis of factors Involvement and Expectations.

Involve the whole environment

Keep in mind that not all interaction, interfaces or tutorial activities need to occur straight in front of the user, but can also involve the whole environment, for a more positive or playful experience. Having an interesting environment and encouraging users to look around, will make them more involved in the experience, which strengthens presence. An example of this is the confetti that rains over the tester at the end of our tests when they finished a puzzle, as this made them feel more physically there and encouraged them to look around at the same time.

This is based on a couple statements regarding several different factors, with focus on Visual realism, Involvement and Shutting out the outside. Witmer and Singer also talks about similar benefits[15].

Fitting but unrealistic elements over realistic ones that stand out

Users tends to accept unrealistic elements in a virtual world more, over realistic ones that stand out or frustrate. This is also true for interactions and interfaces, so it’s not as simple as just making everything as realistic as possible. One should rather make sure not to create friction between the user and the environment, or to make the user stop in their tracks and reflect over how the virtual world actually works, since this could break presence.

This is supported by our findings about factor Physical realism in the factor analysis, but also ties in a bit with Negative emotions.

Putting pressure on a user can increase presence

Putting pressure on a user, or placing them in an urgent situation, can increase presence, as long as you avoid frustration and it's not too much for the user. It's a delicate balance, though, so this would need some extensive testing. A variant of this could be to make a tutorial task really challenging, in order to get them more focused on it.

A couple statements about the factor Negative emotions led to this conclusion, and is also supported by a study by Riches et al.[32]. This could for example be one of the engaging triggers mentioned about the factor Involvement, as well.

Avoid tasks that users struggles with a lot

When a user fails to solve a task or puzzle, because it's too hard to solve, or just because they do not have enough information to solve it, frustration or the feeling that they are not there but just playing an "ordinary game", is a common response, which then breaks presence.

This is related to our conclusions about factors Negative emotions and Immersion, and is also connected to, if not directly supported by, a study by Slater and Steed[30].

Irritation or distractions leads to loss of attention

When a user gets irritated or distracted by something, they are not as attentive on an interface or tutorial as they would be. They might also lose interest or attention if a task is too small or goes too quick to solve, since they therefore will not have the time to get involved in it, or look for instructions.

A couple of testers talked about this effect, which is somewhat supported by our general analysis of the factor Negative emotions. Other related factors in the factor analysis include Involvement and Interface awareness. Some studies also support the general ideas of this guideline[75, 30].

7.3.2.5 A final note on factor importance

In Table 7.5 we can see the number of guidelines that each factor in the factor analysis (see Section 7.3.1) contribute to. The most prevalent among these presence factors are Interface awareness and Negative emotions, shortly followed by Expectations and Involvement. Of course, Interface awareness is closely connected to tutorials and interfaces, as well as being something that the guidelines cover a lot, so that seems like something worth to consider. We'd also like to raise Expectations as important to prioritize, since user expectations are hard to know and needs some work to get right, especially since it more or less affects the whole experience. The importance of Negative emotions is two-fold, since it might both break or enhance presence, depending on its nature. Lastly, Involvement will come in handy when used creatively, to capture user attention and keeping them engaged.

Body realism	0
Control	3
Diegetic sound	2
Expectations	5
Immersion	2
Interface awareness	7
Involvement	5
Negative emotions	6
Physical realism	2
Shutting out the outside	3
Visual realism	2

Table 7.5: How many virtual tutorial design guidelines that each factor in the factor analysis contributed to.

8

Discussion

In this chapter we discuss how the process went according to our plans and any unforeseen obstacles during the project like the Covid-19 pandemic. We also discuss our results and what seems to be interesting, either by strengthening our claims or being a counter to it. Finally we look at what could be further worked on with these prototypes and what research could follow this study.

8.1 Discussion of Process

Generally we decided to split up as much work as possible, to be able to work on different things at the same time and then later merge them together. Since both of us have similar backgrounds in programming and design work, with some differences in 2D and 3D design tools, this meant that we could easily jump into whatever item that had the highest priority.

This parallel way of working generally worked very well and meant that we got a lot accomplished in a short amount of time, but the process could probably have been improved upon further. For example, this could have been done by dividing it into even smaller chunks of work in order to better divide the workload and make it easier to merge the work together in the end.

8.1.1 Pre-study

During the pre-study (see Section 6.1), a lot of the research went into studying up on presence and how to test for it. This was necessary for us to do this kind of project, but it took a lot of time away from studying interactions and how to make tutorials work in VR specifically. There was never time for a more extensive literature review and analysis (eg. *Systematic Literature Review*[31] or *Systematic Mapping*[49]), although this would probably have been helpful, should there have been time for it.

For example, we could have spent more energy on researching tutorial design. We do have a sense of what makes a good tutorial from playing games, however, and are aware of the general discussion about them. A good tutorial is something that you don't really think about, but helps teach you the mechanics of a game. This is somewhat in contrast with what we wanted to test, but a more thorough study on tutorials could possible inform the design process of the tutorials.

The game study where we tested VR games to get an understanding about how they worked with tutorials and interfaces in general, helped a lot with inspiration about how to think about the different designs, especially the flat design Tutorial A and Tutorial B. It also helped us with inspiration for the puzzles, because our limited experience in VR before starting the project meant that we did not really know the conventions used in VR when it comes to for example controls and how to interact with objects, and so on.

8.1.2 Design

The design process was perhaps too short. Since we didn't have much experience in developing for VR, we knew that we had to allocate as much time as possible to the development phase and that time was mainly taken from the design process. Although the development surely needed that allocated time, and the design process at least covered the basics, it did feel a bit rushed and excluded things like exploring different concepts before deciding on one, testing different prototypes with users prior to development, and other useful design methods.

Out of the design methods we did use, though, it worked well with us ideating and sketching on a whiteboard and really collaborate closely through our different thoughts and combining them into a coherent design. This allowed each of us to influence the design and gave both a sense of ownership.

The derived presence factors (see Section 3.1.2) also really helped guide the design process of Tutorial B and C, since these were attempts to improve on the flat base concept of Tutorial A, and we did this through the use of these factors. The different games we played in the pre-study was perhaps the most influential in guiding the design of Tutorial A, although we also had general tutorial design from ordinary non-VR games in mind for that. However, these design decisions were mostly not more than loosely based on previous research, which they perhaps could've been if the designing phase of the project would have been longer.

Later during development, after testing the puzzle ourselves, we had a hard time reading the symbols on the cube from certain angles, so to help with readability, we added different colors that light up in the symbols. This was a trade-off between how realistic it looks and how usable it is, although this specific design decision was not mentioned by any of the testers.

For such a short and rushed design phase, the resulting puzzle and game design was more than adequate for our purposes, so all in all, it could have gone worse.

8.1.3 Development

Starting the development phase with building the environment and where the player would sit and how the environment around it should work, meant that we could use the same environment for all puzzles and then fit the puzzles into the environment.

On the other hand, this meant that we did not develop the most important parts of the experience first, which perhaps could have given us greater freedom in that aspect. This is hard to know, of course, and the resulting virtual environment and puzzles still ended up quite good.

Finding free assets for the environment and sounds, instead of modelling everything ourselves, was crucial to be able develop the prototypes in such a short amount of time. The assets for the table, puzzle cubes and tutorial backgrounds still had to be made from scratch by us, though, which of course took time from the rest of the development, but it also gave us a lot of freedom with their looks and functionality. In fact, finding free pre-made assets that fit our rather specific requirements for these parts just might have taken more time than it took for us to model them ourselves, anyway.

A difficult interaction to solve without breaking presence in VR was holding one object in both hands. Early on, we decided that the best solution for this would be to not do it and rather swap hands when trying to use both of them. However, this instead lead to it being almost impossible to interact with Puzzle 2 like a regular Rubik's cube. After a some thought and testing, we decided to abstract the interaction into pressing the trigger that we felt mirrored the action in the real world, if you would use your index finger to rotate a side. This turned out to be very unintuitive and broke a lot of users' presence, but on the other hand, according to the questionnaire data, we could see that Puzzle 2 had the highest overall presence score.

If we would have held off on the decision to make the Puzzle 2 interaction more abstract, and been better prepared for how to develop these interactions in VR, we could maybe have made a puzzle that felt more intuitive and possibly even one that didn't break presence. One could argue that maybe we should have chosen a different kind of puzzle instead, where the interactions wasn't as tricky in VR. At the same time, a lot of users told us that the puzzle itself really took advantage of the capabilities of VR since you had to look around it the whole time when you rotated the sides. With enough time and testing it could probably have been a really good puzzle, but as it is now, it stands out from the others and is a point where the presence can break.

Issues like the interaction of Puzzle 2 could perhaps have been easier solved, had we been able to use pilot tests on the prototypes. Pilot tests would also have generated valuable feedback on the usability of both the puzzles and tutorials, that surely could have smoothed out a couple of obvious problems that we didn't notice until development was over and testing was already in progress. The Covid-19 pandemic was part of the reason for us excluded this, but we could surely have found ways to work around this, which we did not.

8.1.4 The effects of the Covid-19 virus pandemic

The largest effect the Covid-19 pandemic had on the process was the testing, since the school locked down during the later stage of the development of the prototypes.

We thought it would not be correct and safe enough to let all testers come and test the prototype and use the same HMD, even if we would clean it and make sure that no one had any symptoms before testing. Since it was possible to test remotely, we decided to do that since it was the safest way to test, even though it's not an optimal way to observe and test. However, one potentially positive thing that came out of this, is that we got to test it in the players' natural space where they usually play. This way we noticed something that might not have been noticeable otherwise, which was the problem when they reached for things in the virtual environment and accidentally hit their physical table. For some, this was a big problem in terms of sense of presence, while others thought we did a good job of having the position and height of the table in the virtual environment about in the same place as they had their own table.

In terms of effect on the development process, the pandemic made it harder to discuss decisions and collaborate. Using different tools such as Discord as a voice chat, we could join each others' voice channels whenever we needed and quickly talk about things. We also had meetings every morning before starting to work where we could get caught up on how the other was doing and what they were going to do that day. This made it easy for us to ask each other questions if we needed help with something, since we had scheduled time for that each day. Besides this, most of the development was done alone, unless we had something concrete to talk about.

When analyzing the data, we decided to divide the interviews between us instead of going over each one of them together. This meant that we were more efficient in terms of time, but the resulting codes only has one perspective and the analysis from them might not be as well thought about. It could have been done together, but it's harder to collaborate on these things when at different locations, and it was easier to do them separately and after that go through what the other did to see if we had anything to add.

In general, the pandemic did affect us a lot, but we are also very aware of that it could have been so much worse. So if anything, we were lucky to actually be able to continue and finish the project despite of it.

8.1.5 Testing

The testing went very well with regard to the situation but it wasn't without it's problems, of course.

By testing remotely only users that had their own HMD could test it, this meant that most users had a lot of experience in VR. However, a few testers had a lot less experience playing in VR than the majority of the testers, which lead to them experiencing more problems with the controls than the others, which could lead to them experiencing less presence.

Since a lot of our testers spoke Swedish, it felt natural to also have the interviews in Swedish. This could have resulted in a difference in understanding the translation of the questionnaire, that was in English. It's possible that this led to a misunderstanding of the questionnaire, since some of the questions was quite vague and hard to understand, and we tried to explain them as well as possible without biasing the results. On the other hand, having the interviews in Swedish might have helped the testers explain and describe their experience, but they still used a lot of the English words like “presence” and “immersion”.

We also got some responses about it being hard to answer the questionnaire without having something to compare it with. A good idea could have been to let them try some other game, beforehand, and answer the questionnaire about that as well. This was difficult to accomplish, though, since the tests were remotely performed and they would have needed that other game as well. However, this was not something that we considered before testing, but rather an afterthought.

As mentioned in Section 8.1.3, some problems that the users experienced during testing, we would have been able to fix before hand if we had done some pilot testing, which we unfortunately was not able to do. One of these problems was that the users were instructed to throw the puzzle cube behind them, but it turned out it was hitting the back of the chair which meant that they had to do it a lot of times before getting it right. Another issue was that it was too much text in the tutorials, something that we surely could have worked more on, had we known it (and not been so engrossed in development).

8.1.6 Data analysis

Before analyzing the data we decided to transcribe everything from the interviews, which helped a lot with the analysis, but was also probably the most time consuming part of the entire process. Maybe we could have taken better notes during the interview and mainly focused on them and only go back to the recorded interview to get the context. This would probably have saved us a lot of time but could also have lead to us missing very important quotes from the interviews. Thankfully, we had planned for this taking a lot of time and had allotted several weeks for testing and transcribing.

When moving into the analysis phase of the project, we had planned to do a *Deductive Content Analysis*[31] first, followed by a custom take on a *Kano Analysis*. However, when we read up on the *Kano Analysis* at this point, to refresh our understanding of it, we realized that we’ve completely misunderstood it and therefore couldn’t use it as we thought we could. On the other hand, we could still do more or less the same thing as we had planned, only without involving *Kano* into it. None the less, this hiccup in the process was definitely not positive for the process, and made us double back to rethink our analysis strategies. It seems to have turned out well in the end, but the question is: could it have been even better?

We kept track of what puzzle and tutorial each quote referred to but this might be misleading in the end, since the interviewee could have been talking about a particular puzzle, but the connected tutorial would then have been counted as well, even though the quote might have had nothing to do with the tutorial in question.

On the other hand, it was extremely helpful that we kept the personal ID-codes and actual quotes alongside the codes, so that we could easily look something up in the transcription, to get the context. The ID-codes were introduced to reduce our personal bias for each test person, so going back into the transcription more or less counteracted this idea, and we don't know if this actually had any impact either way.

From the emerging codes, "Difference in tutorials", was introduced to distinguish the quotes where the testers were talking about the difference between the different tutorials, but this was mostly not the main thing that the quote was about. This helped us keep better track of this specific detail when writing the summations and what to think about when designing tutorials for VR.

Early on in the project we had an idea that presence could be connected to the concept of flow-state, but we decided that it was out of the scope of this study, although something that could be studied further. The emerging code "Flow-state" comes from this idea and that some users said that they were so focused on what they were doing and felt more present in the virtual world because of this. Sometimes the code "Involvement" was used instead, since it was hard to tell the difference between them from these quotes. A high sense of presence seems to be something that users can experience when they are in a flow-state, so how these two are connected and if one causes the other is something that could be very interesting to study further, even if we decided not to.

Generally, though, the emerging codes we saw during the *Deductive Content Analysis* did not contribute a lot, and in the end, we decided to exclude them when summarizing the qualitative results.

8.2 Discussion of results

In this section, we discuss the results and their generalizability, covering first the quantitative and then the qualitative results.

8.2.1 Quantitative result

The quantitative data was not collected to draw conclusions from, but rather to give context and contrast to the qualitative data from the interviews. The quantitative data generated from the interviews mainly consist of the amount of times we used certain codes (see Appendix B for more details on this), but it's rather arbitrary and depends a lot on our own interpretations of the quotes. Moreover, there might be some things that the interviewees were very vocal about but in a larger context didn't have such a big impact.

According to the quantitative results from the questionnaire, Tutorial B had the lowest overall score while Tutorial C had the highest (see Table 7.2). We speculated beforehand that Tutorial C would induce the most presence, and Tutorial A would have the lowest. One reason for this difference in results could be that the testers didn't notice Tutorial A as much as the others, because it was very similar to a "regular" tutorial and therefore it did not break their presence. Almost all of the testers said that they originally did not notice the differences between these tutorials. This might mean that they just did not consciously think about it, but that it's still had a subconscious effect.

Another thing that stood out, was that the "Experienced Realism" sub-scale in the questionnaire data scored generally much lower than the others, regardless of how we looked at it. This could be because of how the prototypes looked and that they were "low poly" and had flat colors, without much detail. It could also stem from the fact that the hands was see-through and purple, instead of looking more like the tester's own human hands.

There was an interesting contrast between the results from the questionnaires to the qualitative data from the interviews and the following analysis. Some of these results somewhat contradict what they said in the interviews, such that Puzzle 2 had the highest presence score while in the interviews a lot of people said it was the prototype they felt least presence in. It's hard to know why it was this way but it could be because the questionnaire could be hard to understand and answer. Another possibility is also that, like they said in the interviews, it still was a very engaging puzzle and when answering the questionnaire that was the thing they were thinking about instead.

We had a hypothesis that the more diegetic the tutorial is the higher the sense of presence would be, since it fits better with the world and the users expectations of it. However, according to the interviews they did not really seem to notice the difference between them, but they still answered in the questionnaire that Tutorial C had the highest presence score. So while this doesn't prove anything, it seems like an interesting venue to keep exploring.

8.2.2 Qualitative result

First off, since this is a qualitative study, none of the resulting conclusions and guidelines are actually proven to work. Neither by having a large enough sample size to make statistically founded conclusions, nor by putting the guidelines to work in a fitting project, to see if they are in fact beneficial. However, all of the guidelines are based on things that several testers agreed upon, and in some cases all of them, besides also mostly having a clear basis in previous research as well. So although not proven, there is hopefully enough value in these guidelines to warrant future use or research.

The factor analysis is of course only us interpreting what each interviewee has said, and even if we base it on quotes, it's still just an interpretation of the meaning of each quote. So we might have misinterpreted some things, and we have not double checked any of our interpretations with each interview, to see if they're correct or not. In a way, everything qualitative is reliant on interpretations and the transformation of information, of course, so it's a problem for all similar research that is hard to avoid, but a problem none the less.

Regarding the guidelines, there have been a selection process where only conclusions that fit the research question of this thesis have been included. This selection has of course been performed by us, who are subjective humans, and not through some kind of mathematical or statistical selection, which perhaps could have been more unbiased. Even though we have strived to remain unbiased ourselves, we are not entirely without it, however hard we try. Which means that important points could have been excluded, or unimportant points included, in the result.

Another aspect of biased selection is that of extrapolating on certain conclusions. While the conclusions themselves are grounded in data, some of the guidelines also suggest further action or thought, as a result. As an example, in guideline "Users do not pay attention to realistic or expected things", we talk about the importance of users feeling secure and confident in the virtual environment. Our extrapolated suggestion is then to introduce new concepts in a safe and secure environment, to not break presence. While this suggestion is not part of our data, there is a clear logical connection between the two. However logical, though, since it's not part of the data, it could actually be entirely untrue. Logical conclusions such as these are present in most of the guidelines.

The derived factors that a big part of this study relies on, are a summary of several different other factors from different studies on the subject. Although these factors have been a great help throughout the project, they were summarized by us in the pre-study of the thesis with the intention to facilitate our work, and are not primarily phrased for the use during VR development. In other words, the factors may not translate well into a game developer setting. As mentioned earlier, though, we did not test our resulting guidelines after their creation, so we do not know either way.

During design and development, it was decided to have a longer tutorial at the start of the first of the three tests that each tester would play through, to make sure that each participant was eased into the controls of the experience. However, this also means that the first of the three tutorials always was a bit different, than the other two, regardless of which was which. This was decided in the interest of fairness, but might actually have affected the study in ways we do not know. For example, the longer, slower first tutorial might have been more friendly to beginners, while expert users might have been irritated by its slow pace. We did not examine the impact of this in particular.

8.3 Future work

There is a lot that can be further worked on in this study. For instance, we would like to try and solve the interactions problems with Puzzle 2, as well as making the tutorials clearer and more understandable. Further developing these prototypes to make an actual game out of it would be very interesting and could also make it possible to make a large scale study if the questionnaire was somehow integrated into the game and distributed to a large population.

A possible continuation of this study could be how presence is affected when interacting with complicated interfaces like a menu or an inventory and apply the same guidelines and factors to see if they are applicable to other interfaces than tutorials. It might be that they are more important in other cases since it is more focused on the actual interface and interactions with it.

Another is to try and develop a tutorial with these guidelines in mind and see how it could affect presence. Does following these guidelines give the tutorial a high level of presence, or could it be a very small effect or no real effect at all? It would be interesting to do a large quantitative study on this to try and statistically prove if the guidelines work, and in that case which.

One could also do the same study, but with a non-VR game, to see the differences in how users see tutorials and experience presence in and outside of VR. It might be very different, since it's not as easy to feel presence in a non-VR game. Or one could take these guidelines and see if they could improve on an experience while not in VR, as well.

As mentioned in Section 8.1.6, we discussed early on that flow-state could be connected with presence but that it was outside of the scope of this study. However, this is something that warrants it's own study, since several users said that they felt most presence when they were fully engaged with a puzzle and that it felt like it was the only thing they focused on.

9

Conclusion

The purpose of this project has been to qualitatively examine what affects presence in VR games when it comes to complicated, content-heavy user interfaces such as tutorials, and to produce valuable insights about it that could help game designers tackling similar problems.

The research question of this project was:

Which factors needs to be considered, and how should these factors be used, to not break presence when creating tutorials for virtual reality games?

To answer this question, the subject was researched and examined, to find out previous notions about what affects presence in VR. This resulted in a list of 11 summarized factors that could influence a sense of presence in users in different ways.

- Body realism
- Control
- Diegetic sound
- Expectations
- Immersion
- Interface awareness
- Involvement
- Negative emotions
- Physical realism
- Shutting out the outside
- Visual realism

A prototype of a VR game was then built (see Figure 9.1), with three different tutorials guiding the user. Each of these tutorials were designed with different approaches in mind, based on our derived presence factors, while the content of them stayed the same. After the prototype was completed, we let a group of user testers try out these prototype tutorials, and evaluated them using the *Igroup Presence Questionnaire* and semi-structured interviews about their experiences.

During this study, the Covid-19 pandemic broke out across the world, which enforced some difficult limitations on this project, mainly prohibiting us from conducting user tests in person. Although we managed to continue with the project, the amount of testers and the testing circumstances still suffered, which might have affected the study negatively.

The resulting data from the interviews and questionnaire was compiled and analyzed with regard to the above mentioned summarized presence factors. This resulted in a large number of different insights and conclusions about how to best enforce and not break the feeling of presence in a user. We then processed all of this to find out what mattered the most, when it comes to tutorials in VR specifically, and ended up with a list of 20 design guidelines for virtual tutorial presence:

- A tutorial should function in the way a user expects it to
- Aim for intuitive controls and interactions
- Design for natural and realistic interfaces
- A strong focus makes the user forget about their surroundings
- Interfaces that does not fit in makes a user more aware of them
- Make all information as clear as possible to avoid confusion
- Favor concrete controls over complicated or abstract ones
- A solid and coherent sound design is important
- Tutorial contents are more important than visual presentation
- Avoid long texts or too much information at once
- A coherent virtual world can be more important than photo-realism
- Use interface sounds with care
- Avoid creating scepticism or uncertainty about the environment
- Get users acclimatized with problematic aspects as soon as possible
- Let users figure out some things by themselves when possible
- Involve the whole environment
- Fitting but unrealistic elements over realistic ones that stand out
- Putting pressure on a user can increase presence
- Avoid tasks that users struggles with a lot

- Irritation or distractions leads to loss of attention

Out of these, a central conclusion was that visual photo-realism were less important than making sure that everything in the virtual environment fit together, and to work with users' expectations, to not break their presence. In line with this, we also noticed that the visual design of the tutorials actually didn't seem to impact much on the users' experience, but the content of them on the other hand did.

Being a qualitative study, the resulting analysis has no statistical ground, and the guidelines have not been tested to be true. This is therefore something that would need further research. Other possibilities include doing a similar study on non-tutorial interfaces, or tutorials outside of VR, to see how general our guidelines are, or if they might be generalized further. The game prototype in itself could also be further developed in order to release it in a more commercial setting.

Figure 9.1: An overview of the resulting prototype.

Bibliography

[1] . "presence in vr tutorials". [Online]. Available: "https://torbjornsson.itch.io/presence-in"

[2] "presence in virtual reality when interacting with non-diegetic graphical user interfaces". [Online]. Available: "https://github.com/Torbjornsson/DATX05-Master_Thesis"

[3] (2018) Celeste. [Online]. Available: http://www.celestegame.com/

[4] (2011) † (The Binding of Isaac official website. [Online]. Available: https://bindingofisaac.com/

[5] (2018) Fast Travel Games | Apex Construct. [Online]. Available: https://www.fasttravelgames.com/apex-construct/

[6] (2018) Creed. [Online]. Available: https://survios.com/creed/

[7] (2019) Fast Travel Games | The Curious Tale of the Stolen Pets. [Online]. Available: https://www.fasttravelgames.com/thecurioustaleofthestolenpets/

[8] (2019) Until You Fall | Schell Games. [Online]. Available: https://www.schellgames.com/games/until-you-fall

[9] (2019) Stress Level Zero. [Online]. Available: http://www.stresslevelzero.com/

[10] (2019) Fast Travel Games | Budget Cuts 2: Mission Insolvency. [Online]. Available: https://www.fasttravelgames.com/budget-cuts-2-mission-insolvency/

[11] (2007) Team Fortress 2. [Online]. Available: https://www.teamfortress.com/

[12] (2008) Dead Space. [Online]. Available: https://www.ea.com/sv-se/games/dead-space/dead-space

[13] 2019: The Year Virtual Reality Gets Real. [Online]. Available: https://www.forbes.com/sites/solrogers/2019/06/21/2019-the-year-virtual-reality-gets-real/#3c37c83a6ba9

[14] M. Slater, “A note on presence terminology,” *Presence connect*, vol. 3, no. 3, pp. 1–5, 2003.

[15] B. G. Witmer and M. J. Singer, "Measuring presence in virtual environments: A presence questionnaire," *Presence*, vol. 7, no. 3, pp. 225–240, 1998.

[16] (2019) Oculus Rift S: VR-headset för VR-klara PC-datorer | Oculus. [Online]. Available: https://www.oculus.com/rift-s/

[17] S. V. G. Cobb, S. Nichols, A. Ramsey, and J. R. Wilson, "Virtual reality-induced symptoms and effects (vrise)," *Presence: Teleoperators and Virtual Environments*, vol. 8, no. 2, pp. 169–186, 1999. [Online]. Available: https://doi.org/10.1162/105474699566152

[18] J. M. Fulvio and B. Rokers, "Variation in visual sensitivity predicts motion sickness in virtual reality," *bioRxiv*, 2020. [Online]. Available: https://www.biorxiv.org/content/early/2020/03/10/488817

[19] (2018) Beat Saber - VR Rhythm Game. [Online]. Available: https://beatsaber.com/

[20] (2016) Google Earth VR. [Online]. Available: https://arvr.google.com/earth/

[21] (1996) The history of Tomb Raider | PC Gamer. [Online]. Available: https://www.pcgamer.com/the-history-of-tomb-raider/

[22] (1980) The Official Site for PAC-MAN - Video Games & More. [Online]. Available: https://pacman.com/en/

[23] M. Lombard and T. Ditton, "At the Heart of It All: The Concept of Presence," *Journal of Computer-Mediated Communication*, vol. 3, no. 2, 09 1997, jCMC321. [Online]. Available: https://doi.org/10.1111/j.1083-6101.1997.tb00072.x

[24] M. V. Sanchez-Vives and M. Slater, "From presence to consciousness through virtual reality," *Nature Reviews Neuroscience*, vol. 6, no. 4, pp. 332–339, 2005.

[25] D. Johnson and J. Wiles, "Effective affective user interface design in games," *Ergonomics*, vol. 46, no. 13-14, pp. 1332–1345, 2003.

[26] J. Schirm, G. Tullius, and J. Habgood, "Towards an objective measure of presence: Examining startle reflexes in a commercial virtual reality game," in *Extended Abstracts of the Annual Symposium on Computer-Human Interaction in Play Companion Extended Abstracts*, 2019, pp. 671–678.

[27] V. Schwind, P. Knierim, N. Haas, and N. Henze, "Using presence questionnaires in virtual reality," in *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, ser. CHI '19. New York, NY, USA: Association for Computing Machinery, 2019. [Online]. Available: https://doi.org/10.1145/3290605.3300590

[28] igroup presence questionnaire (IPQ) overview | igroup.org - project consortium. [Online]. Available: http://www.igroup.org/pq/ipq/index.php

[29] M. Usoh, E. Catena, S. Arman, and M. Slater, "Using presence questionnaires in reality," *Presence: Teleoperators and Virtual Environments*, vol. 9, no. 5, pp. 497–503, 2000. [Online]. Available: https://doi.org/10.1162/105474600566989

[30] M. Slater and A. Steed, "A virtual presence counter," *Presence: Teleoperators & Virtual Environments*, vol. 9, no. 5, pp. 413–434, 2000.

[31] B. Hanington and B. Martin, *Universal methods of design.* Rockport Publishers, 2012.

[32] S. Riches, S. Elghany, P. Garety, M. Rus-Calafell, and L. Valmaggia, "Factors affecting sense of presence in a virtual reality social environment: A qualitative study," *Cyberpsychology, Behavior, and Social Networking*, vol. 22, no. 4, pp. 288–292, 2019.

[33] M. Slater and S. Wilbur, "A framework for immersive virtual environments (five): Speculations on the role of presence in virtual environments," *Presence: Teleoperators & Virtual Environments*, vol. 6, no. 6, pp. 603–616, 1997.

[34] S. Nichols, C. Haldane, and J. R. Wilson, "Measurement of presence and its consequences in virtual environments," *International Journal of Human-Computer Studies*, vol. 52, no. 3, pp. 471–491, 2000.

[35] D. Nunez, "Effects of non-diegetic information on presence: A content manipulation experiment," *Annual International Workshop on PRESENCE*, pp. 19–25, 2007.

[36] S. W. Bindman, L. M. Castaneda, M. Scanlon, and A. Cechony, "Am i a bunny? the impact of high and low immersion platforms and viewers' perceptions of role on presence, narrative engagement, and empathy during an animated 360° video," in *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, ser. CHI '18. New York, NY, USA: Association for Computing Machinery, 2018, p. 1–11. [Online]. Available: https://doi.org/10.1145/3173574.3174031

[37] J. Frommel, K. Fahlbusch, J. Brich, and M. Weber, "The effects of context-sensitive tutorials in virtual reality games," in *Proceedings of the Annual Symposium on Computer-Human Interaction in Play*, 2017, pp. 367–375.

[38] M. Troxler, S. Qurashi, D. Tjon, H. Gao, and L. Rombout, "The virtual hero: The influence of narrative on affect and presence in a vr game," in *Affective Computing and Context Awareness in Ambient Intelligence*, G. Nalepa, Ed. CEUR Workshop Proceedings, 2018, affective Computing and Context Awareness in Ambient Intelligence
, AfCAI ; Conference date: 19-04-2018 Through 20-04-2018. [Online]. Available: http://afcai18.webs.upv.es/index.html

[39] K. Brooks, "There is nothing virtual about immersion: Narrative immersion for vr and other interfaces," *alumni. media. mit. edu/˜ brooks/storybiz/immersiveNotVirtual. pdf (accessed May 2007)*, 2003.

[40] M. Csikszentmihalyi, *Finding Flow: The Psychology Of Engagement With Everyday Life*. Basic Books, 2020. [Online]. Available: https://books.google.se/books?id=niWwDwAAQBAJ

[41] J. Takatalo, "Presence and flow in virtual environments: An explorative study," *Master's thesis, University of Helsinki*, 2002.

[42] E. Fagerholt and M. Lorentzon, "Beyond the hud-user interfaces for increased player immersion in fps games," Master's thesis, Chalmers University of Technology, 2009.

[43] I. Iacovides, A. Cox, R. Kennedy, P. Cairns, and C. Jennett, "Removing the hud: The impact of non-diegetic game elements and expertise on player involvement," in *Proceedings of the 2015 Annual Symposium on Computer-Human Interaction in Play*, ser. CHI PLAY '15. New York, NY, USA: Association for Computing Machinery, 2015, p. 13–22. [Online]. Available: https://doi.org/10.1145/2793107.2793120

[44] P. Salomoni, C. Prandi, M. Roccetti, L. Casanova, and L. Marchetti, "Assessing the efficacy of a diegetic game interface with oculus rift," in *2016 13th IEEE Annual Consumer Communications Networking Conference (CCNC)*, Jan 2016, pp. 387–392.

[45] P. Salomoni, C. Prandi, M. Roccetti, L. Casanova, L. Marchetti, and G. Marfia, "Diegetic user interfaces for virtual environments with hmds: a user experience study with oculus rift," *Journal on Multimodal User Interfaces*, vol. 11, no. 2, pp. 173–184, Jun 2017. [Online]. Available: https://doi.org/10.1007/s12193-016-0236-5

[46] A. Shannon, A. Boyce, C. Gadwal, and T. Barnes, "Effective practices in game tutorial systems," *Foundations of Digital Games*, 2013.

[47] E. Andersen, E. O'Rourke, Y.-E. Liu, R. Snider, J. Lowdermilk, D. Truong, S. Cooper, and Z. Popovic, "The impact of tutorials on games of varying complexity," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2012, pp. 59–68.

[48] J. Rowley and F. Slack, "Conducting a literature review," *Management research news*, vol. 27, no. 6, pp. 31–39, 2004.

[49] K. Petersen, S. Vakkalanka, and L. Kuzniarz, "Guidelines for conducting systematic mapping studies in software engineering: An update," *Information and Software Technology*, vol. 64, pp. 1 – 18, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0950584915000646

[50] Unreal Engine | The most powerful real-time 3D creation platform. [Online]. Available: https://www.unrealengine.com

[51] Unity Real-Time Development Platform | 3D, 2D VR & AR Visualization. [Online]. Available: https://unity.com/

[52] Visual Studio Code. [Online]. Available: https://code.visualstudio.com/

[53] Rider: The Cross-Platform .NET IDE from JetBrains. [Online]. Available: https://www.jetbrains.com/rider/

[54] Git. [Online]. Available: https://git-scm.com/

[55] GitHub. [Online]. Available: https://github.com/

[56] Agile Alliance. [Online]. Available: https://www.agilealliance.org/

[57] What is Scrum? | Agile Alliance. [Online]. Available: https://www.agilealliance.org/glossary/scrum/

[58] What is a Kanban board? | Agile Alliance. [Online]. Available: https://www.agilealliance.org/glossary/kanban-board

[59] M. Mohseni, B. Liebold, and D. Pietschmann, *Game Research Methods [Lankoski & Björk].* ETC Press, 04 2015.

[60] (2015) Keep Talking and Nobody Explodes - Defuse a bomb with your friends. [Online]. Available: https://keeptalkinggame.com/

[61] (2016) Obduction. [Online]. Available: https://obduction.com/

[62] (2013) Race The Sun | Flippfly. [Online]. Available: http://flippfly.com/racethesun/

[63] (2016) No Man's Sky. [Online]. Available: https://www.nomanssky.com/

[64] blender.org - Home of the Blender project - Free and Open 3D Creation Software. [Online]. Available: https://www.blender.org/

[65] (1975) About Rubik's Cube. [Online]. Available: https://www.rubiks.com/en-us/about

[66] Google Hangouts. [Online]. Available: https://hangouts.google.com/

[67] Google Forms. [Online]. Available: https://www.google.com/intl/en/forms/about/

[68] B. Vector". "low poly cliff pack". [Online]. Available: "https://assetstore.unity.com/packages/3d/environments/landscapes/low-poly-cliff-pack-67289"

[69] "Rad-Coders". "low poly: Foliage". [Online]. Available: "https://assetstore.unity.com/packages/3d/vegetation/low-poly-foliage-66638"

[70] A. Bringer". "simplistic low poly nature". [Online]. Available: "https://assetstore.unity.com/packages/3d/environments/simplistic-low-poly-nature-93894"

[71] B. Vector". Low poly rock pack. [Online]. Available: "https://assetstore.unity.com/packages/3d/environments/low-poly-rock-pack-57874"

[72] K. at Fertile Soil Productions". "modular temple pack". [Online]. Available: "https://fertile-soil-productions.itch.io/temple-modular-collection"

[73] S. Pepwuper". "simple helvetica". [Online]. Available: "https://assetstore.unity.com/packages/tools/gui/simple-helvetica-2925"

[74] B. Su". "v-light volumetric lights". [Online]. Available: "V-LightVolumetricLights"

[75] A. Sutcliffe, *Multimedia and virtual reality: designing multisensory user interfaces.* Psychology Press, 2003.

A

VR game review, extended

For each of the following games, a brief description and then analysis with focus on graphical user interfaces and sense of presence.

Beat Saber

The concept of this music game is very simple: the player holds two light-sabers in each hand while glowing boxes fly towards them and loud dance music play. The point of the game is then to hit the boxes synchronized with the music, and failing to keep the rhythm will end a level in failure.

GUI in the game is flat and put statically on the sides, so that you do not generally notice it. When you miss a note, a specific sound will play together with a visual text queue, so you do not need to have time for reading the text. The fast pace and dance music really draws you in, and a rather large sense of presence was experienced.

Apex Construct

Set in a twisted version of Swedish capital Stockholm, this game is about a lost human taken under the wings of an ominous AI with the name of Fathr, who needs help fighting another AI called Mothr. In the game, one will explore the city, fighting off robots with a bow, while trying to help Fathr.

Tutorial and GUI elements are mostly flat, white text or symbols hanging in the air, and combined with sometimes complicated or buggy interfaces, these things tend to break the feeling of presence. On the other hand, the game world is immersive and fighting robots was quite involving and stressful, which works in the opposite way in terms of presence. Some of the diegetic elements in this game felt a bit bulky and awkward, and therefore not as presence-inducing as we would have though, which is not always the case, and this gives the impression that it's not always as simple as just making something diegetic.

The Curious Tale of the Stolen Pets

A game directed towards a young audience, where you stay stationary while you visit different doll-house-like worlds with tiny animal inhabitants. Each world is filled with small puzzles where you interact with the environment, while a framing story is told by your supposed grandfather.

The static nature of the gameplay, the soothing sounds and music, combined with the fact that almost all GUI and interaction in the game is diegetic, all works towards a great level of presence. This is despite the fact that the graphics are in a cartoon style, rather than trying to be realistic, which is an interesting point to keep in mind.

Budget Cuts 2

Continuing after the first game, Budget Cuts 2 is a humorous story of a human who, together with the whole of humanity, is about to be optimized out of existence by vicious robots. Your task is to sneak, stab and shoot (with bow and arrow) through the corporation of TransCorp to put an end to it.

While being similar to Apex Construct in play style, most diegetic elements in Budget Cuts 2 works better and enforces presence more. Generally, the game is rather polished, which vaguely felt like it helped presence as well. The beginning of the game started out quite hard, though, and had to be replayed several times before progressing, and this repetition and the frustration that followed, really broke presence.

Google Earth VR

Technically not a game, this tool is more of an exploratory experience, based on the desktop application with the same name. You can fly around the Earth at different heights, zoom in on famous cities or just visit your old childhood home, with the flick of your thumb. Granted, the 3D-environment of the whole of Earth does not have very high fidelity, but it's complete enough to be an immersive VR experience.

Since this experience is based on collected footage from Google and other private sources, there is a big difference in fidelity between different places or different height from the ground. At some places, the difference is clearly visible, and this really broke presence. On the other hand, surprisingly enough, very low fidelity was not a huge issue in itself.

Keep Talking and Nobody Explodes

An older game that has been refitted for the VR experience. This one is completely collaborative, where the person wearing the VR headset has a bomb in their hands that they're trying to disarm, while getting expert help from anybody else in the room, who cannot see the bomb but instead has access to a large bomb manual with detailed instructions. It's a stressful but fun cooperative experience.

Since this game was adapted to VR after launch, it's quite obvious that this was the case, mostly because interaction with menus are constantly awkward: each meny is an object that you have to pick up with one hand and then interact with, with your other hand. Since these interactions are not optimal, it breaks presence. On the other hand, the very stressful nature of the game and the dark room in which you try to disarm the bomb, is all very involving.

Obduction

Another game where the VR support has been added afterwards. As new as it is, it still follows the same kind of gameplay as its predecessors Myst and Riven, as a point-and-click adventure where you try to solve a mystery in a deserted alien world, into which orbs of Earth has been magically transported.

The three-dimensional world is decent enough to not be a problem in VR. However, all GUI elements are a mess, and it's obvious that this is because the developer did not spend enough time on this when converting the game to VR. The most obvious problems are actual graphical bugs, but there are also issues with FMV (Full Motion Video), as these are flat and not volumetric, and therefore does not feel like real characters at all – despite being actual recordings of real people.

Boneworks

A recent VR game that tries to push the boundaries of what is physically possible to stand in VR without getting VR sickness. It's about a company that has created a virtual world with "void energy", which unstable nature is posing a problem, while at the same time handling an AI gone rogue. However, the study never got past the very long tutorial intro.

Moving about in this game is exclusively by locomotion, in comparison to most other first person VR games that work with teleportation. Generally, locomotion is known to induce more nausea or VR sickness than other means of travel, and sure enough, a stronger sense of vertigo was present in the beginning of this game. However, high fidelity and a general consistency when it comes to interaction with objects, still seems to give a good sense of presence. And after a while, some of the VR sickness went away as the player adapted, which is interesting in itself. There were also some non-diegetic elements that perhaps didn't help presence, but used sparingly was at least not distracting.

No Man's Sky

Since this game was released (not in VR) it has seen a lot of updates and changes from its developer, and one of those has been the addition of a rudimentary VR-mode. The game is about a person waking up on a strange planet, their ship and equipment damaged, having to survive in this strange hostile environment. The game soon opens up and lets you explore different planets, solar systems and even galaxies.

Another example that a conversion of a non-VR game into VR after it's released can be hard and produce problematic results. In this game, the whole HUD stays in one place and totally counteracts the possibility to turn physically in place. There are also different cut-scene moments when the camera angle shifts, which is jarring in VR. When piloting your ship in its cockpit, there are some nice diegetic interfaces, but they are hard to read and understand. Generally, this VR-adaption works rather bad in terms of presence.

Until You Fall

A polished rogue-like in VR where you play as a knight come back from the dead, guided by a mysterious voice who wants you to defeat an ancient evil that has infested the world. You have one weapon in each hand and perform a series of slashes and blocks on each enemy, while also trying to duck away from larger attacks. Every time you die, as is usual in rogue-likes, you can upgrade or expand your gear and thus change playing styles between runs.

This game contains a lot of interfaces and statistical information to keep track of, and so on. As a consequence, this game is cluttered with half-way diegetic floating interfaces in different places. As a game and rogue-like, this works very well, and the game is very enjoyable. However, the experience come off as arcade-like and the sense of presence is low.

Race The Sun

Yet a game that was not originally made for VR, but converted to it later. The concept is very simple but imaginative: you possess a very fast vehicle and the purpose of the game is to keep up with a setting sun in front of you. In your way will also be a lot of static or moving obstacles, as well as power-ups that give you more time or points, and moving through shadow lowers the sun at greater speed.

In-game, the presence was high, almost only because of the high, stressful tempo, combined with a general lack of user interface. And although the player travels in incredible speed forward, trying to avoid obstacles, for some reason it's not a VR-sickness inducing problem, perhaps because when you sit down on a chair, it felt like all obstacles moved towards you, rather than the opposite. The menus in this game, on the other hand, were a real mess and destroyed what presence there was, whenever they showed up.

Creed: Rise To Glory

A classic boxing game, in the style of old NES game Punch Out!!, where the goal is to start out low, train with legendary Rocky Balboa and rise in the ranks of boxers until you are the best boxer in the world. The mechanic is simple: you block and punch by moving your controllers in the same manner as in real boxing. The game is physically taxing in a way that a play session is like a light work-out.

When you stand stationary, trying to punch your opponent, the presence is high, as you behave almost exactly as you would have in a real fight. When you need to move, though, the mechanic of swinging your arms at your side while standing still, to move forward, does not feel natural but instead kind of hilarious. It's an interesting technique of locomotion, and might work well in the long run, but laughing about it broke presence as it was.

B

Data from coded interviews, extended

As described in Section 6.5 the interviews were transcribed and then the quotes that were relevant to presence and the study was coded. In Table B.1, the number of times each code was used and at what priority position the original factors (see Section 3.1.2) together with the added codes (*Acclimation, Flow-state, Difference in tutorials* and *Difficult to understand*) was the codes used for the analysis. In Figure B.1 we can see that the Interface Awareness code is very frequent in first priority and that is very understandable since the test was about the interface and a lot of questions was about the interface and whether they saw the difference between them.

Codes	Priority 1	Priority 2	Priority 3	Total
Acclimation	3	4	4	11
Body realism	9	2	0	11
Controls	37	21	7	65
Diegetic sounds	15	2	1	18
Difference in tutorials	7	21	4	32
Difficult to understand	0	3	0	3
Expectations	16	33	12	61
Flow-state	0	2	2	4
Immersion	15	28	14	57
Interface awareness	53	24	9	86
Involvement	38	24	9	71
Negative emotions	8	16	4	28
Physical realism	23	15	10	48
Shutting out the outside	15	27	7	49
Visual realism	28	15	6	49

Table B.1: How many times the code was used and at what place

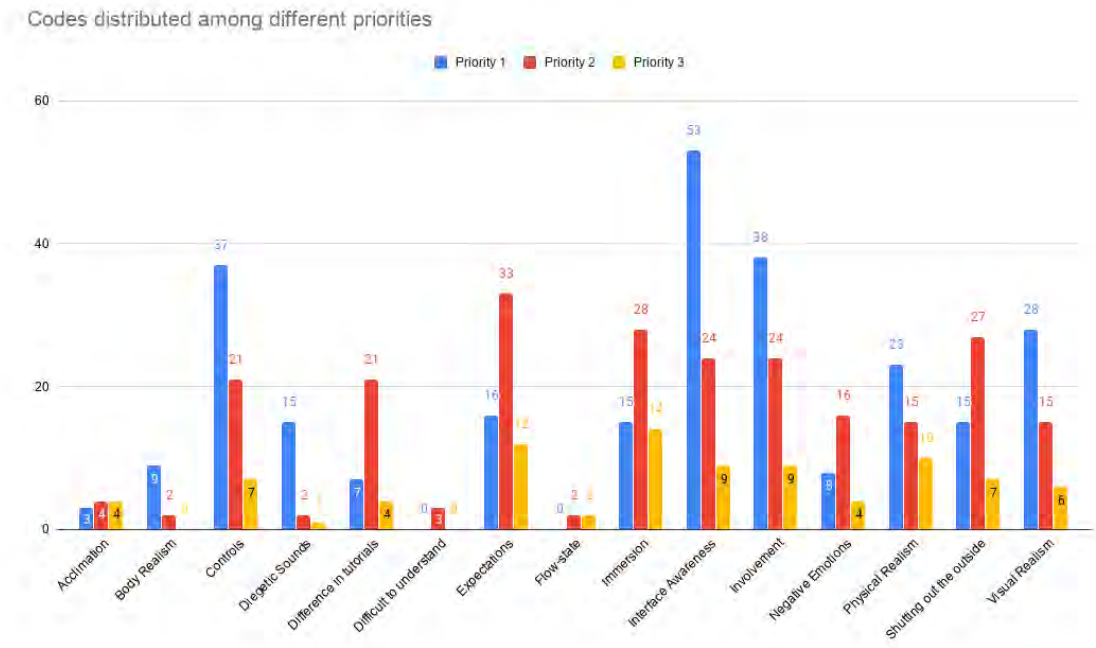


Figure B.1: Chart over codes distributed among different priorities

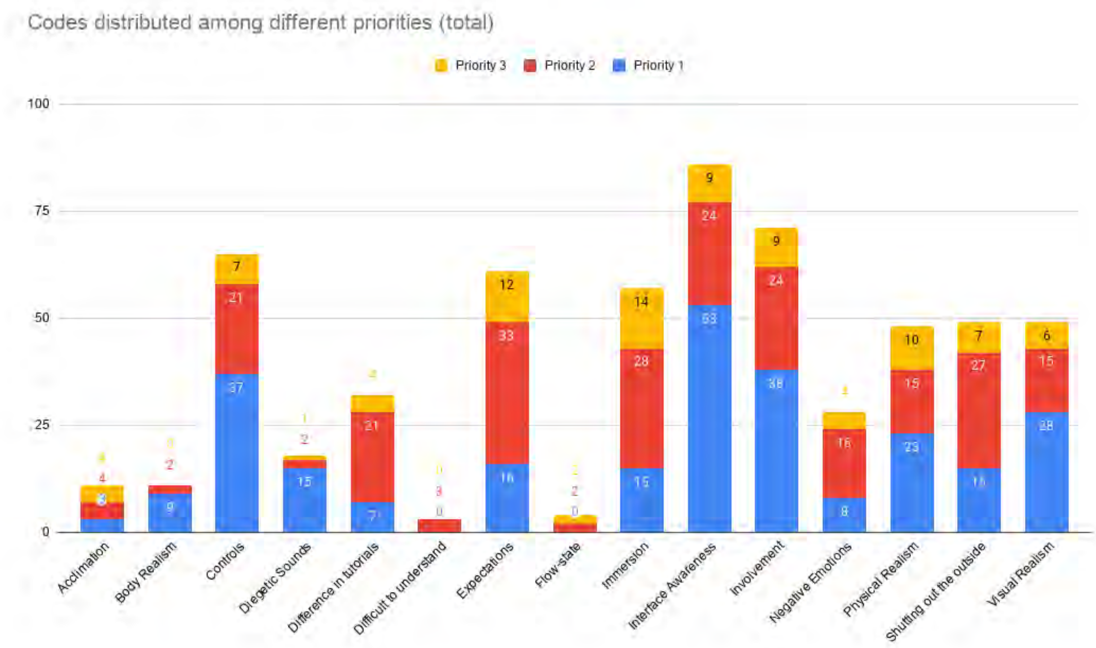


Figure B.2: Chart over the total distribution of codes

As seen in Table B.2, the number of times each code was used in a particular puzzle or in general in all puzzles (there are some that are used in 2 of the puzzles, or it was not clear to what puzzle they where talking about). Here we can see that some codes are more common as a general theme across all puzzles while others are more common in a certain puzzle. The number of times each puzzle was mentioned can be seen in Figure B.4, and Puzzle 2 is talked about the most, while Puzzle 3 is mentioned the least, but the difference between them is not that big.

Codes	Puzzle 1	Puzzle 2	Puzzle 3	All puzzles
Acclimation	4	2	2	1
Body realism	1	1	2	5
Controls	15	28	5	7
Diegetic sounds	2	3	1	12
Difference in tutorials	5	2	4	19
Difficult to understand	1	1	1	0
Expectations	15	17	12	15
Flow-state	0	1	1	2
Immersion	11	11	10	17
Interface awareness	12	22	14	27
Involvement	13	10	9	29
Negative emotions	2	8	7	7
Physical realism	17	9	5	14
Shutting out the outside	6	7	6	24
Visual realism	3	5	3	34

Table B.2: How many times the code was used for each puzzle

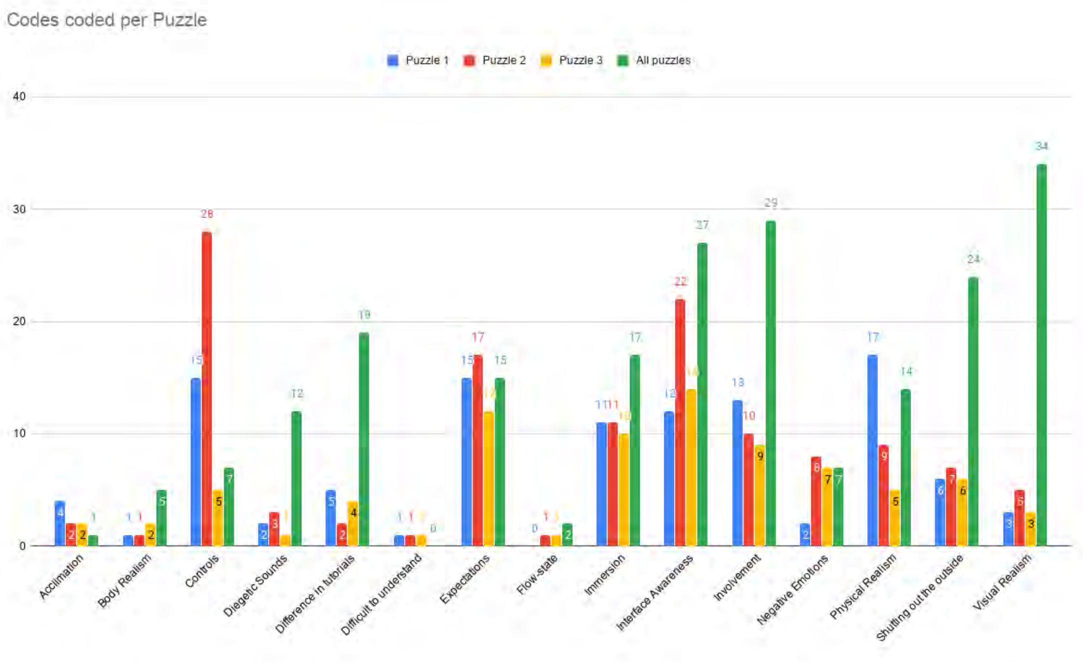


Figure B.3: Codes used on the different puzzles

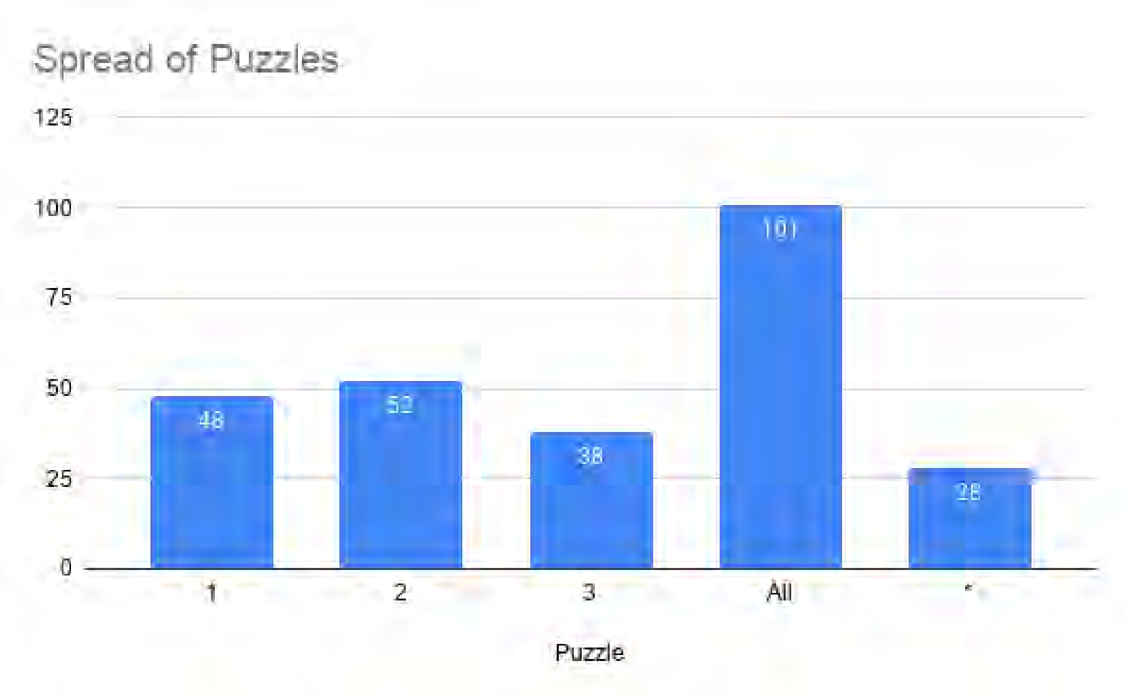


Figure B.4: How many times the code was used for each puzzle

With the codes for the tutorials in Table B.3 the number for all tutorials is the same as in all puzzles since this is when the quote was general and not in any specific puzzle or tutorial. Some tutorials were mentioned and coded more than others (see Figure B.6). For example, Tutorial C is used more than twice as much as Tutorial A. This is probably because we asked questions regarding the difference in diegesis between the tutorials and Tutorial C was very different from the other two and Tutorial A was the closest to the flat tutorial screens that the testers might be used to in other games.

Codes	Tutorial A	Tutorial B	Tutorial C	All tutorials
Acclimation	1	6	1	1
Body realism	0	2	2	5
Controls	9	16	23	7
Diegetic sounds	0	1	5	12
Difference in tutorials	1	2	8	19
Difficult to understand	1	1	1	0
Expectations	8	14	22	15
Flow-state	0	0	2	2
Immersion	7	11	14	17
Interface awareness	13	8	27	27
Involvement	8	15	9	29
Negative emotions	4	8	5	7
Physical realism	5	14	12	14
Shutting out the outside	6	5	8	24
Visual realism	2	2	7	34

Table B.3: How many times the code was used for each tutorial

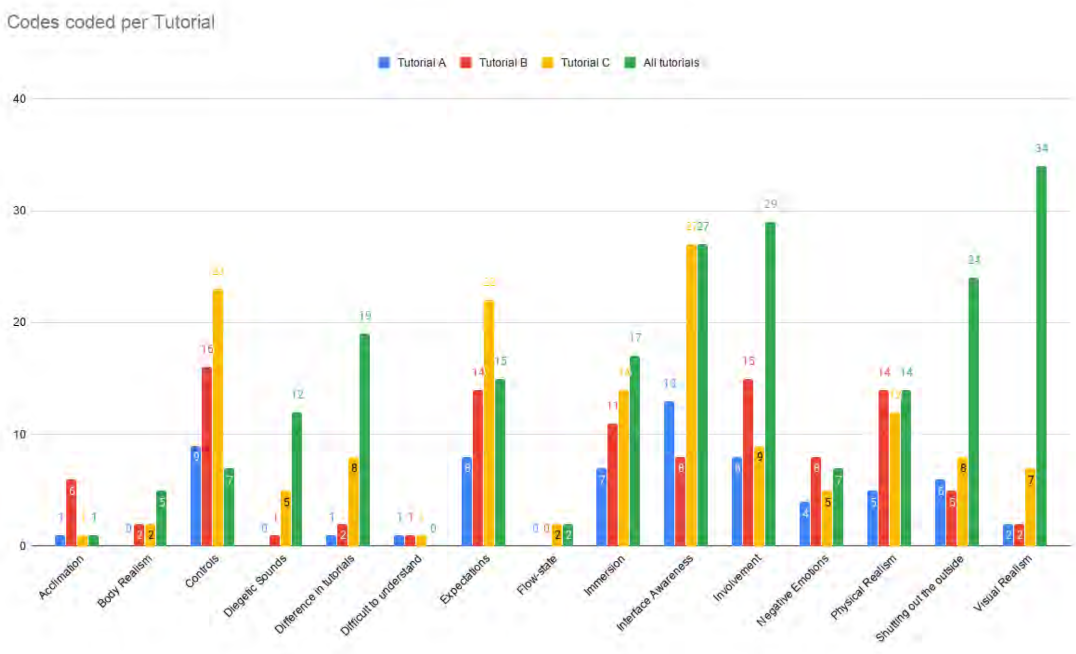


Figure B.5: Codes used on the different tutorials

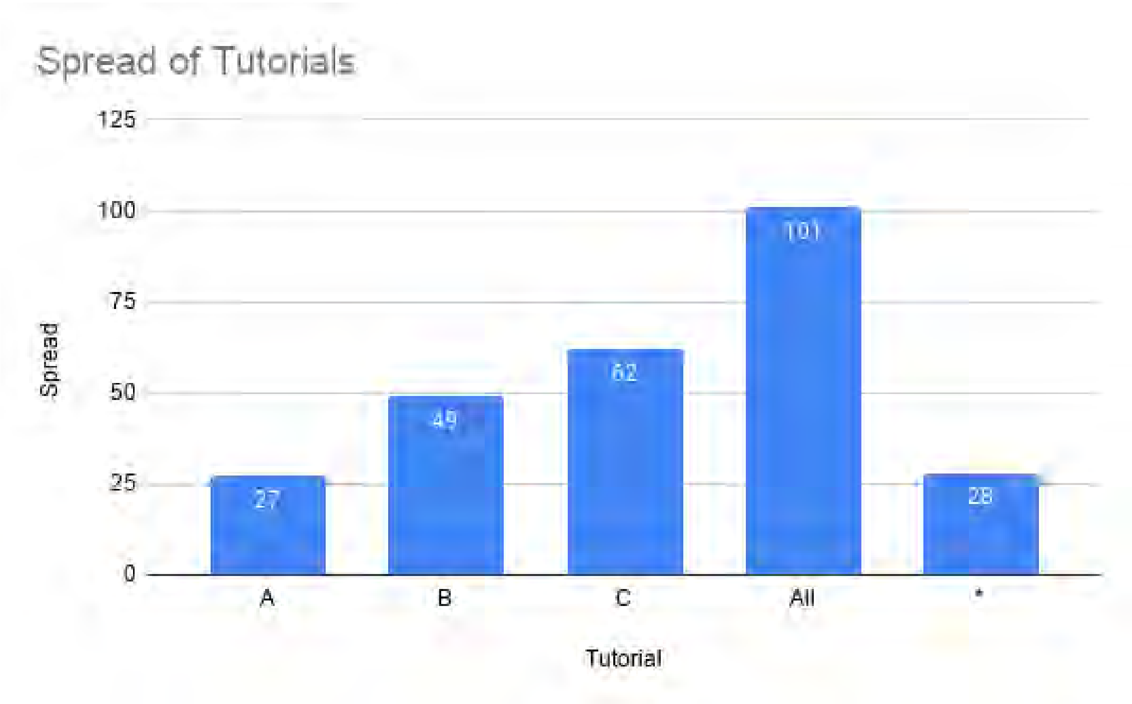


Figure B.6: The number of times the tutorials was mentioned in quotes

C

Consent form

Presence in VR playtest - Consent form

This consent form regards playtests performed in an Interaction Design and Technologies Master Thesis: "Presence when interacting with Graphical User Interfaces in Virtual Reality". When signing this form, you consent to the following:

- I voluntarily agree to participate in this research study.
- I understand that even if I agree to participate now, I can withdraw at any time or refuse to answer any question without any consequences of any kind.
- I understand that I can withdraw permission to use data from my interview within two weeks after the interview, in which case the material will be deleted.
- I have had the purpose and nature of the study explained to me in writing and I have had the opportunity to ask questions about the study.
- I understand that participation involves:
- Playing through three prototypes in Virtual Reality, about 15 minutes each
- Answering a questionnaires about each of the prototypes, three in total
- Participating in a short interview afterwards
- I understand that I will not benefit directly from participating in this research.
- I agree to my interview being audio-recorded.
- I understand that all information I provide for this study will be treated confidentially.
- I understand that in any report on the results of this research my identity will remain anonymous. This will be done by changing my name and disguising any details of my interview which may reveal my identity or the identity of people I speak about.
- I understand that disguised extracts from my interview may be quoted in a Master thesis, conference presentation or published papers on the subject.
- I understand that signed consent forms and original audio recordings will be retained in a private hard drive, only accessible by the researchers and their academic supervisor until the conclusion of the study.
- I understand that a transcript of my interview in which all identifying information has been removed will be retained for two years from the date of the exam board.
- I understand that under freedom of information legalisation I am entitled to access the information I have provided at any time while it is in storage as specified above.
- I understand that I am free to contact any of the people involved in the research to seek further clarification and information.

Researchers:
Robin Lilius-Lundmark, Master thesis student, Chalmers University of Technology
robin.lilius@gmail.com
David Torbjörnsson, Master thesis student, Chalmers University of Technology
david.torbjornsson@gmail.com

Academic supervisor:
Thommy Eriksson, PhD, Director of Master's programme Interaction Design and Technologies, Chalmers University of Technology
thommy@chalmers.se

*Obligatorisk

Name: *

Ditt svar

Do you consent to this? *

- Yes
- No

Skicka

D

Igroup presence questionnaire

Presence Questionnaire

Thanks for participating in this virtual experience, now there are some questions regarding your experience. Please answer from your experience only on this playtest. Some of these questions are very similiar to each other but are necessary for statistical reasons so please answer all of them to the best of your ability.

*Obligatorisk

How long did you interact with the virtual world? (in minutes) *

Ditt svar

Which prototype did you try? (Name of the file) *

Ditt svar

How aware were you of the real world surrounding while navigating in the virtual world? (i.e. sounds, room temperature, other people, etc.)? *

	1	2	3	4	5	6	7	
Extremely aware	○	○	○	○	○	○	○	Not aware at all

How real did the virtual world seem to you? *

	1	2	3	4	5	6	7	
Completely real	○	○	○	○	○	○	○	Not real at all

I had a sense of acting in the virtual space, rather than operating something from outside. *

	1	2	3	4	5	6	7	
Fully disagree	○	○	○	○	○	○	○	Fully agree

How much did your experience in the virtual environment seem consistent with your real world experience? *

	1	2	3	4	5	6	7	
Not consistent	○	○	○	○	○	○	○	Very consistent

How real did the virtual world seem to you? *

	1	2	3	4	5	6	7	
About as real as an imagined world	○	○	○	○	○	○	○	Indistinguishable from the real world

I did not feel present in the virtual space *

	1	2	3	4	5	6	7	
Did not feel	○	○	○	○	○	○	○	Felt present

I was not aware of my real environment. *

	1	2	3	4	5	6	7	
Fully disagree	○	○	○	○	○	○	○	Fully agree

In the computer generated world I had a sense of "being there" *

	1	2	3	4	5	6	7	
Not at all	○	○	○	○	○	○	○	Very much

Somehow I felt that the virtual world surrounded me. *

	1	2	3	4	5	6	7	
Fully disagree	○	○	○	○	○	○	○	Fully agree

I felt present in the virtual space. *

	1	2	3	4	5	6	7	
Fully disagree	○	○	○	○	○	○	○	Fully agree

I still paid attention to the real environment. *

	1	2	3	4	5	6	7	
Fully disagree	○	○	○	○	○	○	○	Fully agree

The virtual world seemed more realistic (not to confuse with photo-realism) than the real world. *

	1	2	3	4	5	6	7	
Fully disagree	○	○	○	○	○	○	○	Fully agree

I felt like I was just perceiving pictures. *

	1	2	3	4	5	6	7	
Fully disagree	○	○	○	○	○	○	○	Fully agree

I was completely captivated by the virtual world. *

	1	2	3	4	5	6	7	
Fully disagree	○	○	○	○	○	○	○	Fully agree

Skicka