Final Report - Draft version

Matthias Tavasszy
4368401

Arjan van Schendel
4366212

Luke Prananta
4288386

Jasper van Esveld
4372581

Bart Ziengs
4391799

June 16, 2016

Preface

tbd

Contents

Introduction and end-users' requirements

Psychosis refers to an abnormal condition of the mind described as involving a "loss of contact with reality". People that suffer from psychosis are described as psychotic. Experiencing this may exhibit some personality changes and thought disorder. Depending on its severity, this may be accompanied by unusual or bizarre behaviour, as well as difficulty with social interaction and impairment in carrying out daily life activities. The non-medicinal treatment of psychosis involves psychological therapies and social support. CleVR, the company this project was conducted for is involved in making those treatments possible by using virtual reality(VR). To mimic a real world setting as realistically as possible, four hardware components are used in this specific case. The full hardware kit consist of an Oculus Rift head-mounted display, a Leap Motion device, a Microsoft Kinect and a set of ManusVR gloves. The problem we were asked to tackle is to integrate these separate components in a seamless way. That is, search for an optimal combination in which they function as one product in its entirety. Either in an synergistic or serendipitous manner.

The approach that was taken is by using so-called user stories that are descriptions consisting of one or more sentences in the everyday or business language of the end user or user of a system that captures what a user does or needs to do as part of his or her job function. At the beginning of the project a lot of initial user stories were composed, that evolved throughout the project. Our primary user stories that reflect it use at best include the following:

- As a user, I want to be able to put my groceries in a shopping cart so that I can take groceries with me.
- As a user I want to be able to pick up items in a physically realistic way so that I can interact with the world in a more natural manner.
- As a user I want to use the oculus rift so that I will be fully immersed in the virtual world
- As a user I don't want my hand to snap between different tracking methods so that I will be more immersed in the virtual world.
- As a user, I want my virtual body to move in a realistic way so that I do not feel disconnected from the virtual body.
- As a developer I want to have a good understanding of the code so that it becomes easier to edit it.

As can be seen, these different stories are formed from the perspective of a certain end-user, mostly the patient that is being treated from psychosis. The stories that are listed above are from the last phase of the project and are evolved from initial stories by making specific enhancements in the requirements that we established during the completion of the project. Every story is, if applicable, divided in one or more sub tasks to specify the needs in an nuanced fashion. An example is given with the user story below that is subsequently divided into two sub-parts. Sub-parts can be seen as direct tasks.

- As a user I want my virtual body to stand on the ground, not hover
  - Make sure that the lowest foot of the Kinect model always touches the ground
  - Fix the hip distort bug when the Kinect loses track of the user.

After this was implemented in the product, this was evaluated with the group where after was decided whether or not this was acceptable and if this needed further modifications. Another example is formed by the task "As a user I want a realistic environment to interact with so that I will get immersed in the world". To verify this we set up an testing day with people that were willing to test it en let them fill in a questionnaire afterwards. Elaboration on testing techniques and procedures will take place in the chapter *Interaction Design and Testing*.

Chapter 1

Product and software overview

The product is a system that uses several pieces of special hardware, controlled by a simulation program that runs on the Unity engine. This program uses the data from the hardware to provide user-computer interaction in a virtual environment. The product also includes a simulation of a supermarket where a person can interact in. The person has the ability to grab and drop virtual grocery items, while also able to slightly move around within the environment. The following hardware is involved in the system:

- Oculus Rift, a head-mounted display (HMD) that provides head-tracking for the system.
- Leap Motion, a small tracking device that can track arms, hands and fingers within the vision of its infrared cameras.
- ManusVR, gloves that can accurately track the bending of fingers.
- Kinect, a device that can track full-body movements within the vision of its cameras.

Figure 1.1: A look at the hardware

In the aspect of software, the simulation is implemented in C-Sharp on the Unity engine. The simulation runs on one computer and all hardware are connected to this computer. The simulation uses the data from the hardware to translate real-life movements to the virtual avatar. The Leap is used to position the hands of the virtual avatar, while the ManusVR is used to correctly bend the fingers. The Kinect provides movement for the rest of the body of the virtual avatar. The Oculus HMD gives the person the ability to look around in the virtual environment. See figure 1.2.

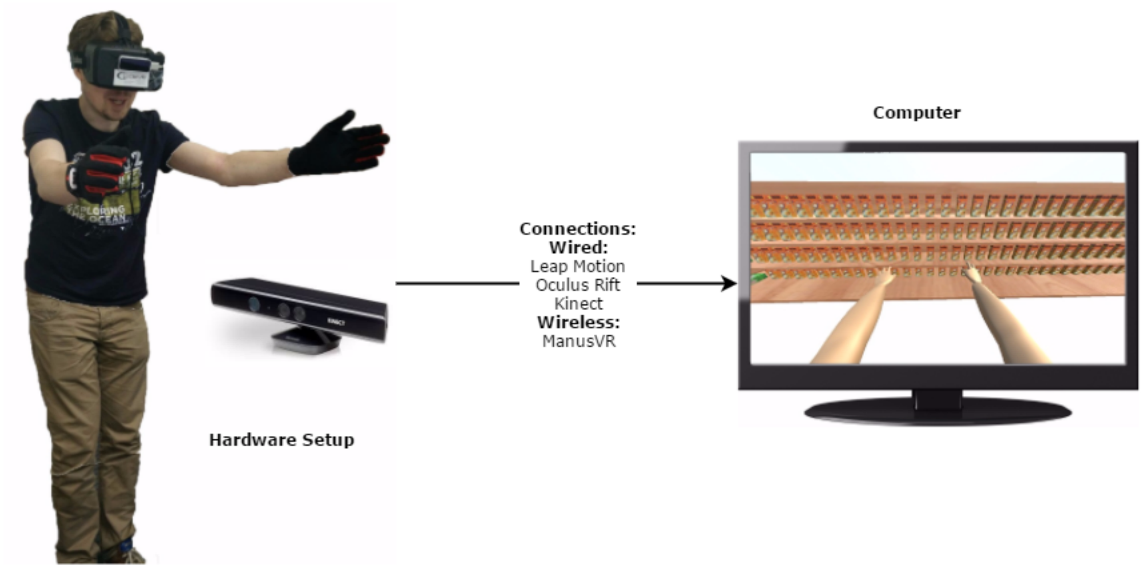

Figure 1.2: A look at the setup

With this the simulation can accurately simulate grabbing motions and actions within a virtual environment, in this case a supermarket environment. The system determines with recognized gestures which grab to apply, a single grab with left or right hand or a double grab that uses both hands. The Kinect also makes it possible to let a person move around within this environment, although to a limited extent.

Chapter 2

Product reflection

When CleVR described the product they had in mind they wanted us to focus on a few software quality characteristics, namely: maintainability, functionality and performance. [2] Performance is important because this can make or break the VR experience. We used Unity's build in profiler to show the performance of our project and this showed that our implementation had a total impact of 2 ms per frame. About 1.9 ms comes from a function within one of the plugins we have to use and we can therefor conclude that performance won't be an issue.
Maintainability was another important aspect, we tried to ensure maintainability by providing documentation for all functions/classes and by implementing several Design Patterns.

2.1 Design Patterns

Design patterns can improve the maintainability of the code, we used three different Design Patterns in our implementation for recognizing grabs. Since there can be multiple grabs on a given moment we implemented the Observer Pattern in which each observer is a grab which gets notified by the subject when it needs to be updated. This subject implements a State Pattern which is used for when the subject holds one or more grabs and for when it does not. The last Design Pattern we implemented is the strategy pattern, this is used for the actual grab itself and defines the strategy for a grab with two hands, just the right hand and just the left hand. Because we used these design patterns editing the code will become allot easier and hopefully help CleVR when implementing it in their own project.

Chapter 3

Process reflection

Here we will discuss problems we encountered during the development process and how we fixed those problems.

3.1 Tooling

The development process has a big impact on the maintainability, quality and functionality of the final product. To help us during the development process we made use of several tools like: StyleCop, GhostDoc, CodeMaid, VSSpellChecker and CloudBuild. Some of these tools we did not use all that often (like CodeMaid) but others proved to be very useful (like StyleCop). In the following subsections we will go over some of these tools (VSSpellChecker is very straight forward and we will therefor not discuss it) and how we used them.

3.1.1 StyleCop

StyleCop is the visual studio equivalent of CheckStyle. Just like CheckStyle it enforces the standards within Csharp for example classes and functions must start with a capital letter. Besides that StyleCop also ensures that comments meet certain requirements like documentation on return values and parameters. We used StyleCop to improve readability during the development process, this not only helped us during the development but will also help CleVR when we deliver the final product.

3.1.2 CodeMaid

CodeMaid is a very handy tool that helps solve StyleCop errors (white space, method order etc.). CodeMaid was recommended by CleVR but we ended up not using it all that much because most of the problems that CodeMaid helped to fix were easy to do by hand.

3.1.3 GhostDoc

To speed up the creation of documentation we used GhostDoc which generates the basic structure for a method or class with a simple shortcut.

3.1.4 CloudBuild

CloudBuild is the Continuous Integration solution for Unity. It works with any Unity project can be connecting with a git repository. For each change within the project it will than automatically build and run all tests resulting in succeeding and failing build. There is however one downside and that is the time for each build when using the free plan. The average build time for our project was more than one hour which meant that using CLoudBuild for things like pull requests became basically impossible.

3.2 Scrum

During the development of our product we followed the Scrum method with sprints of one week. While we have used this method before we still came across some things we could improve. For example at the start of the project we had some problems selecting clear user stories and assigning concrete tasks. We ended up with vague tasks that would be assigned to multiple people. Another problem we had was with our priorities. These would be given correctly but we did not give an explanation to how they came to be. After the feedback we received during the meetings with our SA we would improve our backlogs each week. With backlogs improving we noticed that making a proper retrospective would become easier. This is mainly because the tasks became clearer and therefor easier to reflect on.

3.2.1 Communication: Group

When we started the project we thought that meeting up each day would not be necessary and we could just use Skype when communication was needed. However we noticed that when we did meet we would all be more productive. After the first two weeks we tried to reserve one of the project rooms of Drebbelweg for most of the week.This way whenever one of us encountered a problem we all would be there to help and when changes in the structure of the code were needed we could directly discus them. We would strongly recommend any project group to meet in person.

3.2.2 Communication: Stakeholders

We could not meet up every week with CleVR (the company we were working for), we did however have weekly meetings with Willem-Paul and René Vennik (SA). The weekly meetings were very useful not only because of the feedback we received on our progress, code and sprint but you also get to take a step back and see what work was performed during the last sprint. Even though we could not always meet up with CleVR we still wanted to be able to show our progress. Therefor we used Slack to share short videos showing our progress of that week.

3.3 Git

Just like in previous projects we made heavy use of pull requests and branching. We tried to keep commits directly to the master to a minimum and use pull requests even for small changes. This made sure that atleast 2 people would check the code that is pushed to the master. We did however have a problem with GitHub's statistics when it comes to lines of code. Since we also needed to push our resources (models, textures, etc.) the lines of code for each person got really skewed. Another thing we learned is to make use of the Issues functionality in GitHub, it turned out to be a great reminder since you cannot directly fix every bug you find.

Chapter 4

Description and Evaluation of the functional modules and entire product

When aiming for a composition into modules, our product can be split up into the three components that are shown in the figure below. The first model, *Entering / exiting the environment* is

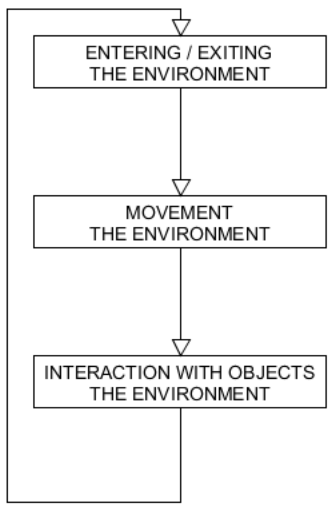

Figure 4.1: Composition of functional modules

just as it says the act of spawning and removing the user in and out of the virtual environment. There is basically no smooth operation for this since there has to be a transition somewhere. Secondly the *movement in the environment* is very limited in the current state of the project since moving around is not possible (yet) due to the missing of proper controls to achieve this. The final functional module, *Interaction with objects in the environment* is the most complex and testable part. This last part is also the most addressed when gaining feedback from the test users. More about this can be found in the chapter about testing and in the chapter on the outlook.

Chapter 5

Interaction Design and Testing

5.1 User and usage situation

The user of this software will use it in a treatment room. This means that there has to be enough space to walk around and make full use of the range of the Kinect. The user is a person that suffers from a psychosis, meaning that he or her sees a distorted version of reality. [1] The entire purpose of this software is to make the user face his or her biggest fear: public places, in particular a supermarket, in virtual reality. In order to make the user show their fear, the virtual world has to look as realistic as possible. This includes a other persons, a realistic supermarket, a working and moving virtual body, and the ability to pick up items. The last two of these were the problems that this software had to tackle.

5.2 Interaction goals

Effective and efficient:
Goal: There should be, close to no latency between real body movements and virtual body movements.
Solution: When tested on the available laptops there was some minor latency. While testing on a more high-end gaming PC, (Intel i5 6600K, MSI GTX 960 4Gb, 16Gb RAM) there was no noticeable delay or latency. The situation in which this system will be used, a PC with at least these specifications will always be available.

Safe:
Goal: All virtual reality movements have to be natural and physically possible, the patient should never be exposed to potentially 'scary' experiences.
Solution: As much impossible body movements as possible were made impossible. For example, the wrists bending the wrong way is impossible.

Utility:
Goal: The user can interact with the virtual world in a natural way, meaning that they can walk around and pick up items.
Solution: The user can pick up items using their hands. Walking around is possible but limited to area covered by the Kinect.

Easy to learn and remember:
Goal: Controlling the system should feel natural, simple, and easy to remember.
Solution: Controlling the virtual body is easy. All a user has to do is move their real body, and the virtual body will move accordingly. Because the movements are the exact same as people use in their whole lives, remembering them is very easy.

5.3 Testing

5.3.1 Method

Users that had to do the tests were put in a room, with enough moving space to be able to comfortably complete the assignments. Firstly, they were asked some questions about their experience and opinion on virtual reality before the test. Then, they put on the ManusVR, and the Oculus with the Leap Motion attached. They were then guided into the correct position in front of the Kinect, then given the following tasks:

1. Look around
2. Look at your hands, move them and your fingers
3. Look at and move your feet
4. Walk towards a shelf
5. Pick up an item with one hand
6. Pick up an item with two hands
7. Throw the item away
8. Put an item on a lower shelf
9. Pick an item up from a low surface (approximately 20 cm from the ground)

Figure 5.1: Guiding a test person to the correct position

When all these tests were done, they were asked briefly about their experience, how easy the tasks were to complete, and what they though were the biggest problems in the system.

5.3.2 Results

Some of the most interesting statistics:

50% of the test subjects were of the age 19-25 years, 12.5% were younger, and 37.5% were older. 75% had never tried virtual reality before, while 94% held a positive attitude towards its

up growth and potential.
When asked how well they though virtual reality would represent reality before the tests, the average answer was 3.1 (on a scale of 1-5), while after the tests, this average slightly decreased to 2.9. On the other hand, the immersion (Did you feel like you were actually in the virtual world?) was rated a 3.6 on average.
The main problem that was mentioned (by 50% of the subjects) was the incorrectness of the tracking. Most of the test subjects mentioned that when they moved their hands too far away from the vision, they would not move correctly, or not move at all.
The system as a whole was, averagely, rated a 7.4. 81.3% really enjoyed doing the tests (rated 5/5).

The full test results can be found in the appendices.

5.3.3 Conclusion

After the tests, and after reviewing the test results, a few things can be concluded. First of all, making a user believe that they are actually in the virtual world does not require perfectly realistic graphics. As one of the users responded : *"Resolution is less important (...) than correct picking up and precise tracking: that disturbs the illusion the most."*
That leads to the following conclusion: it is really hard to correct the hardware's limitations. The Leap Motion for example, has a very limited field of view. When tracking is lost, the system tries to use the data from the Kinect to keep the simulation realistic. The problem is, firstly, that the Kinect is way less accurate than the Leap, so it is easy for the user to notice this difference. Secondly, the transition from one tracking device to the other is not as smooth as it could potentially be, but this can be solved through future developments.

Chapter 6

Outlook

The product as is will not be directly applicable for use within the company. The first reason causing this is that CleVR will use our implementation in their own virtual environments instead of the ones we created and used for testing the product. The flexibility and compatibility was aimed to be high during the project. However, implementation in a whole new environment will simply require some effort.

As stated before in the chapter on testing, half of the persons testing the product indicated incorrectness of tracking as a problem so that is definitely an area in which modifications are necessary. The product works now and the goals that were set prior to this project are reached but this is yet not ready for full usage with people that suffer from psychosis. After these minor bug are fixed, the outlook will be flourishing.

Another aspect has to be looked into is the setting in which it will be ran. If the project in its end state would be built and exported to a runnable file, certain things do not run smoothly together. Also, modifications on the way the product runs can currently just be made by adjusting preferences in unity or adding some code. It might speak for itself that this is not how it is wanted to be upon completion. A final thing worth mentioning is the level of consistency and support. The product has to go through a sufficient amount of testing procedures before it can be marked as reliable. On top of that agreements have to be made between the end-user (i.e. therapist) and CleVR.

Bibliography

[1] Oliver Freudenreich. Differential diagnosis of psychotic symptoms: medical mimics. *Psychiatric Times*, 27(12):56–61, 2010.

[2] Joost Visser. Building maintainable software. 2015.

Appendix A

Testing survey results

The VR tests were held after question 7.

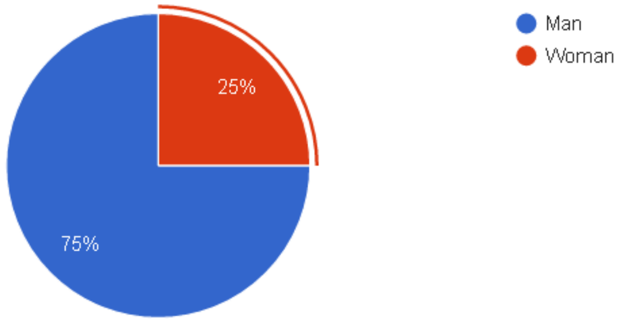

Figure A.1: Are you male or female?

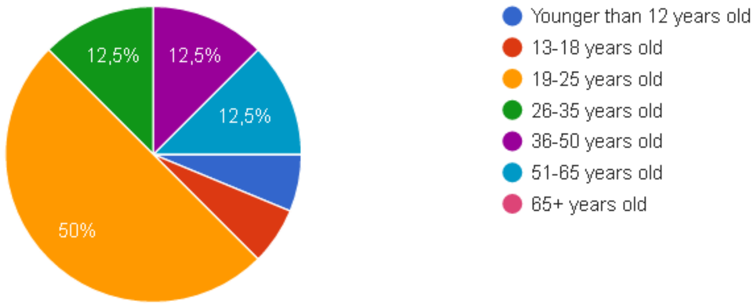

Figure A.2: What is your age?

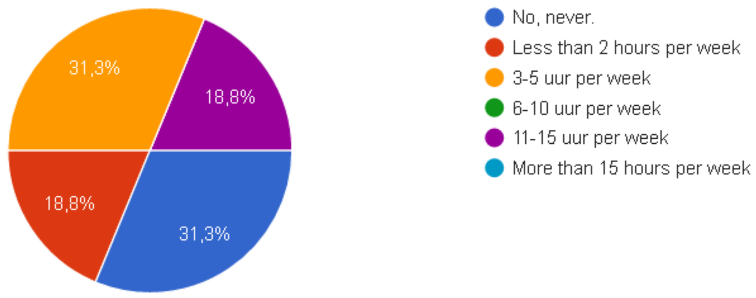

Figure A.3: How often do you play video games?

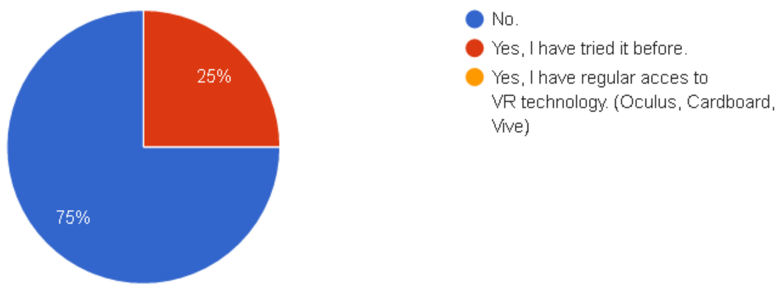

Figure A.4: Have you ever tried virtual reality before??

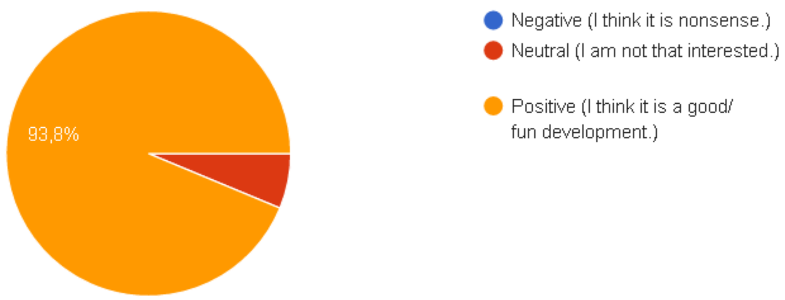

Figure A.5: What is your opinion on virtual reality?

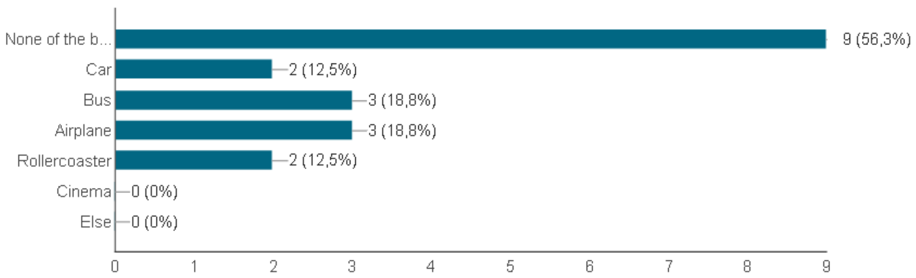

Figure A.6: Do you ever feel sick in one of these situations?

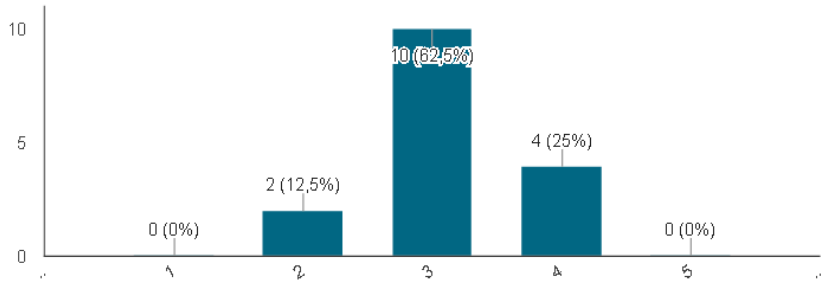

Figure A.7: How well do you think virtual reality will represent the real world?

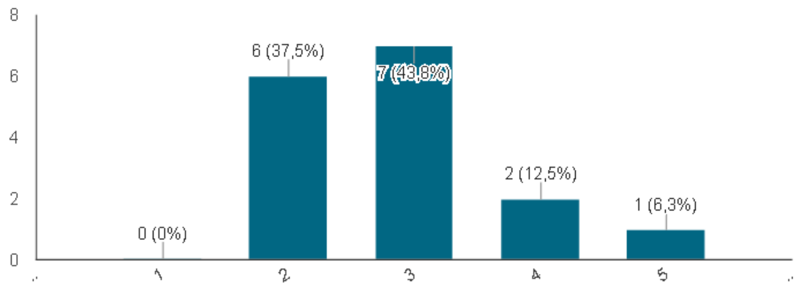

Figure A.8: How well did virtual reality represent the real world?

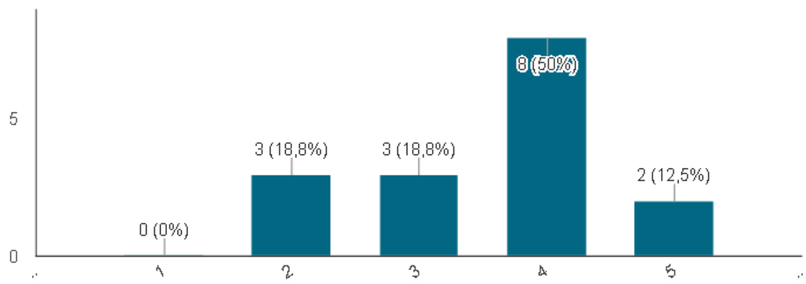

Figure A.9: During the simulation, I felt like I was really in the virtual world.

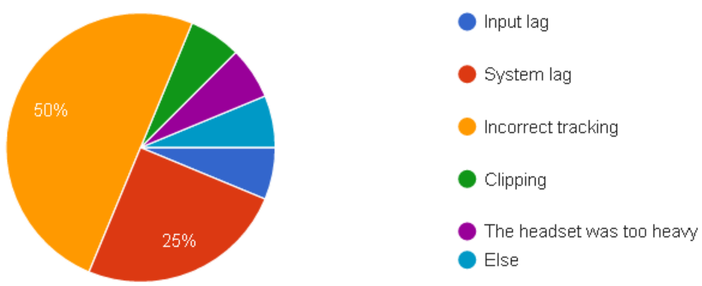

Figure A.10: What was the biggest problem you encountered during the tests?

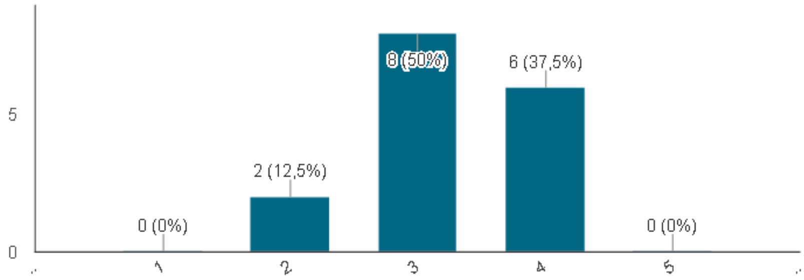

Figure A.11: How easy is it to pick up objects?

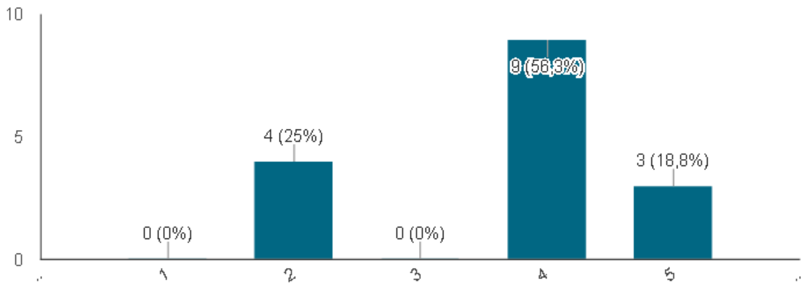

Figure A.12: How easy is it to drop objects?

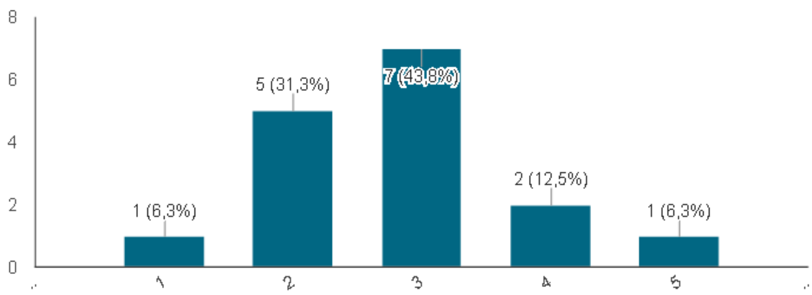

Figure A.13: How realistic are the physics in the virtual world?

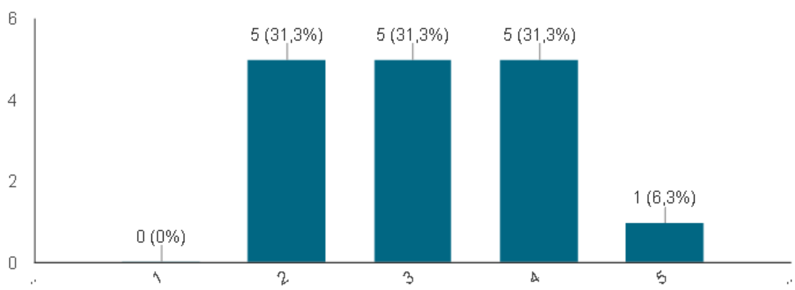

Figure A.14: The environment looks realistic.

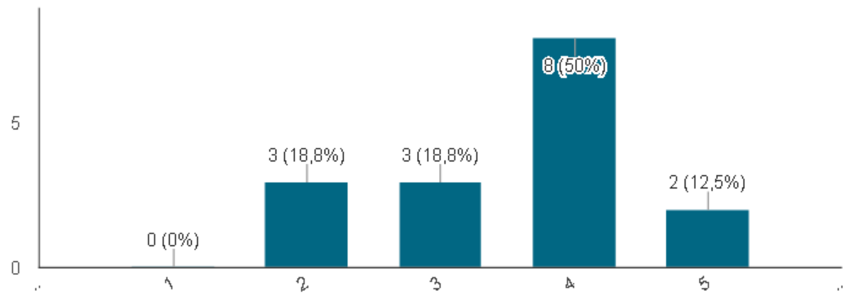

Figure A.15: The objects in the environment look realistic.

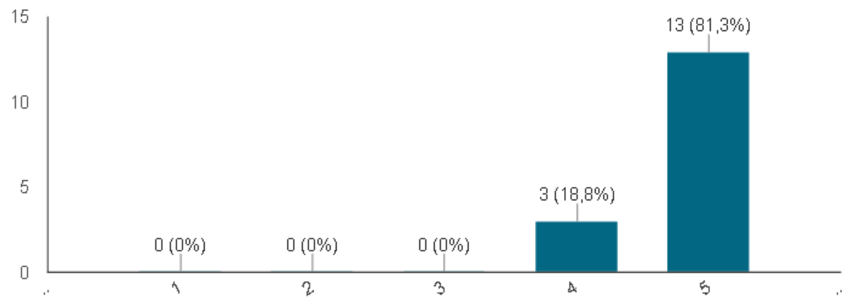

Figure A.16: I enjoyed trying virtual reality.

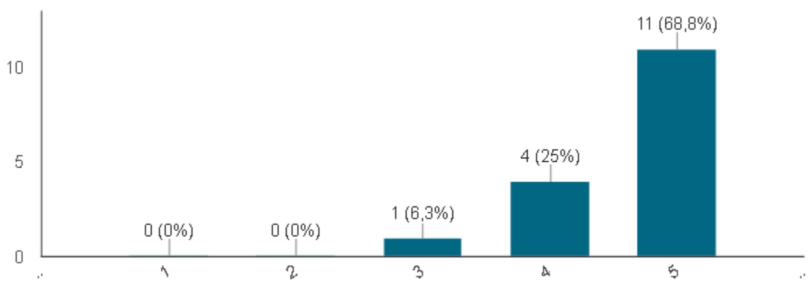

Figure A.17: I want to use virtual reality again.

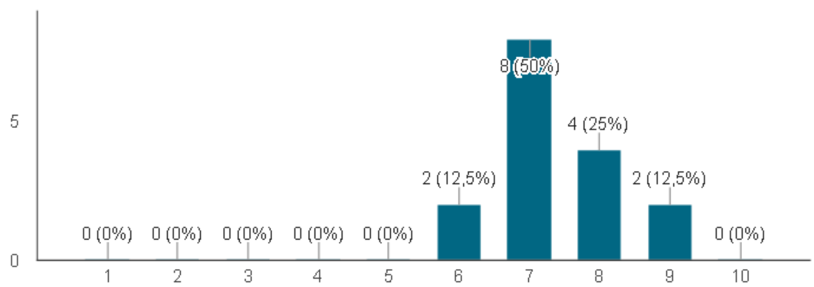

Figure A.18: How would you rate the entire system?