

Creating a Virtual Reality Application for Transporting Items

Written by:

Tjebbe P. Treub

Supervised by:

Dr. J. Zwiers

Critically supervised by:

Dr. M. Theune

Creative Technology BSc

July 2021

UNIVERSITY OF TWENTE.

Abstract

A VR application has been developed to check if items of modular apartments can be transported inside vacant buildings for the 'Doos in doos' project of Innovatiehub Salland. In addition to this, the application also provides a training mode for construction workers and a mode to visualize configurations of apartments inside a building. The successful creation of this application was achieved by implementing goal-based learning, intuitive controls, a simple user interface and methods for the prevention of cybersickness. The application was developed through multiple iterations and co-design with the client. The application is evaluated to have a good to excellent usability score on the SUS test but needs to be further tested on end-users.

Preface

This thesis concludes my bachelor Creative Technology at the University of Twente. The final VR application was created with the purpose of helping Innovatiehub Salland check if vacant buildings are suitable for the 'Doos in doos' project. This thesis has been written and conducted from February 2021 to July 2021.

I would like to express my gratitude to my supervisor Dr. J. Zwiers for his supervision, support and enthusiasm as well as my critical observer Dr. Mariët Theune for her valuable feedback and insights. I would also like to thank Jelle Smith for his pleasant and fun cooperation during this project. I would also like to thank everyone from Innovatiehub Salland for providing this graduation project and the great support. Finally, a special thanks to my family and friends for their feedback and supporting me through this period and in special Michelle Sudjito for providing her car that made it possible to visit the client frequently.

Tjebbe Treub
Enschede, July 2nd 2021

Contents

1	Introduction	4
2	State of the Art	6
2.1	Suitability of Virtual Reality for learning	6
2.2	Learning in Virtual Reality	7
2.3	Creating a Virtual Reality Application	8
2.4	User Interaction in Virtual Reality	10
2.5	User Interfaces for Virtual Reality	11
2.6	Conclusion	13
3	Method	14
3.1	Research Method	14
4	Ideation and Requirement Capture	17
4.1	Co-design	17
4.2	Initial Conceptualisation	18
4.3	User Scenario	20
5	Specification	22
5.1	Constraints on Technical Design	22
5.2	Prototypes	22
5.3	Co-Design	24
6	Realisation	26
6.1	Technical Design choices	26
6.2	Controls	26
6.3	Modes	27
6.4	Do it Mode	29
6.5	Graphical User Interface	30
6.6	User Testing	32
7	Evaluation	35
7.1	Requirements vs Implementation	35
7.2	Feedback from User Testing	36
7.3	Feedback from the client	38

8	Conclusion and Discussion	39
8.1	Conclusion	39
8.2	Discussion	40
8.3	Recommendations	41
9	Appendices	48
9.1	Appendix A	48
9.2	Appendix B	50
9.3	Appendix C	56
9.4	Appendix D	58

Chapter 1

Introduction

Real estate in the Netherlands has two problems. The first problem is the lack of real estate for starters on the housing market, the average search duration for starters is over a year [1]. The second problem is that more and more buildings in the Netherlands are going unoccupied. An example is that in 2022 an expected vacancy of shops in city centres in the Netherlands is estimated at 40% [2].

A new idea to tackle these problems is introduced by Innovatiehub Salland called “Doos in Doos project”. The companies related to this project want to develop real estate for starters located in vacant buildings. Within these vacant buildings, new apartments will be built that facilitate all the needs of starters. The apartments will consist of modular parts. These modular parts give the residents the option to customise certain parts of their apartment to their preferences. The second advantage of using modular parts is that apartments can easily be built up. Building these modular apartments within existing buildings is a new way of construction and thus brings new problems [3]. One of the main problems of this new way of construction is to find out if it is possible to transport parts into existing buildings with relative ease and without having to deconstruct existing structures.

Virtual reality (VR) shows great promise to explore the possibilities of visualizing the transport of items. Next to that, VR is becoming more popular in a lot of different industries. VR is forecasted to deliver strong growth in the upcoming years [4] [5]. Different studies have shown that VR training platforms boost the efficiency of the participants [6] [7]. For these reasons, the usage of VR was chosen to solve this problem.

Therefore the main objective of this research was to create a VR application that is able to indicate if items can be transported into a vacant building. To achieve the main objective, sub-questions have been created that needed to be researched:

1. *How to effectively learn to transport items in VR?*
2. *How to develop a VR application for transporting items?*
3. *How to create user-friendly interaction in VR?*
4. *What is the reaction of the potential end-user(s) on the created application?*

Building on the findings of these sub-questions, a VR application has been created with co-design and multiple iterations based on user testings. This application can fulfil the main goal and in addition to this two extra modes were created. One mode was created for convincing the municipality by visualizing vacant buildings with customizable apartment configurations. The second was created to training construction workers for becoming familiar with the moving process of a building. A visualization of the fitting process of an item is depicted in figure 1.1.

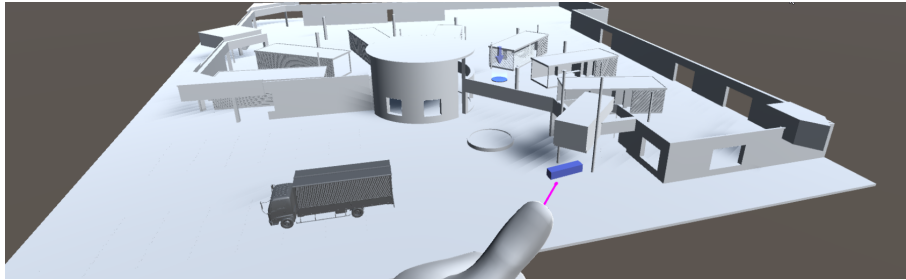


Figure 1.1: Visualization of a fitting process of a building.

Chapter 2

State of the Art

In this chapter, related research is described on relevant topics for this thesis. Existing literature and research will be considered to research how to develop a VR application for transporting items. This chapter describes the findings on the suitability of virtual reality, how to effectively learn in VR and how to effectively develop an application with good user interaction.

2.1 Suitability of Virtual Reality for learning

To validate that virtual reality is a good way of visualizing this problem, it is researched when VR can be applied to what topics and what the disadvantages are. Winn [8] states that “Immersive VR furnishes first-person non-symbolic experiences that are specifically designed to help students learn material.” This is supported by Duncan *et al.* [9], where they conclude that Virtual Learning Environments (VLEs) provide learning opportunities by doing experiments or by constructing knowledge. Pantelidis [10], as cited in Pantelidis [11], concluded from his study that VR could be considered to be applied when a simulation could be used. Next to that Mantovani [12], as cited in Pantelidis [11], named the potential of VR as “Offering the possibility for learning to be tailored to learner’s characteristics and needs.” Horne and Thompson [13] believe that VR can be a useful tool for demonstrating specific processes in built environment education, this is supported by Hilfert and König [14]. On top of that Mantovani [12], as cited in Horne and Thompson [13] indicates that “the point is no more to establish whether VR is useful or not for education; the focus is instead on understanding how to design and use VR to support the learning process”. With this and the other studies supporting this claim, it is suggested that VR can be used for the goal of this research.

VR is not perfect and also has disadvantages. Since VR requires computers and sometimes an internet connection, one of these can create lag or crashes as written by Duncan *et al.* [9]. This research was performed in 2012 and since then the industry related to computers and internet connection has improved,

reducing this issue a lot. Virtual reality is not advised to be used by Pantelidis [10], as cited in Pantelidis [11] if the costs exceed the expected learning outcome. Next to that, he describes that the integration of VR can be a disadvantage. This integration as a new technology did improve over the years as he predicted with new technology and thus is less of a disadvantage. The rest of this list that is presented by Pantelidis [11] does not apply to the goals of this research.

2.2 Learning in Virtual Reality

There are a lot of different ways of learning. Different studies will be compared to find a learning method that will teach the users the most regarding transporting items. According to Schank *et al.* [15] “there is only one effective way to teach someone anything, and that is to let them do it.” This theory of learning by doing was formed by John Dewey in 1922 [16]. Based on this work Schank *et al.* [15] came up with the concept of learning by doing. This theory was applied in the research by Büyüktaşkapu *et al.* [17] and affirmed the importance of learning by doing. Part of these concepts of learning by doing is discovery learning, proposed by Bruner [18]. Discovery learning states that students need to explore a problem on their own before a teacher provides them guidance. Bot *et al.* [19] came up with goal-based scenarios as part of learning by doing. In their approach they let students work towards a goal and let teachers mentor them when getting stuck. Virtual teachers can be created but can not help on a personal level as teachers are able in real life. To fill this gap of guidance a fitting alternative needs to be found. The theory of Bot *et al.* [19] was applied by Vescan [20] and underlined the importance of learning by doing. Stuchlíková *et al.* [21] affirm the great future of creative learning in VR but also make clear that it has practical experiments limits. VR training can not substitute real training in many cases, since real training is still more effective.

Giving instructions in VR could fill the gap of missing a virtual teacher. A way of doing this was recently described by Wolfartsberger and Niedermayr [22] named authoring-by-doing. This concept can be used in training scenarios to communicate the knowledge. When instructions need to be given, a ‘ghost’ appears. This ‘ghost’ is a transparent animation that shows how something should be performed in an ideal way. An example of a ghost in a racing game can be seen in figure 2.1. This could instruct the users when they are stuck in reaching their goal.

Another way of keeping users engaged and improve their learning is by using gamification. This was proven by a literature review performed by Hamari *et al.* [24]. Gamification is a way of adding elements of gameplay to an application. They do note that this is strongly related to the context in which the gamification is implemented. The context of education/learning did consider gamification mostly positive. Gamification also can have negative outcomes such as increased competition. Domínguez *et al.* [25] found that some students did not like the gamification for the extra competition it creates between students.



Figure 2.1: Example of a ghost in Formula E racing game [23].

2.3 Creating a Virtual Reality Application

There are many ways to develop a VR application and this can make it difficult to know where to start [26]. Multiple studies that developed models on how to develop VR applications will be reviewed to find an efficient way for the development process. Pantelidis [11] alleges to have made a model to determine when VR can be used in education and training courses. Critically observing this model it is considered more a guidance model than a model to determine when to use virtual reality. The model shows how to go within ten steps from the requirements to the final application. A second model for the design of VR learning environments was created by Vergara *et al.* [27]. This study is in line with the previous model and builds further on it. The flowchart of the model, depicted in figure 2.2, shows the different steps that are explained to follow and iterate on. They make a distinction in the different levels of user interaction and control, hardware devices and software for programming. They divide the level of user interaction into three levels: passive, exploratory and interactive. Depending on the level of interaction they show different options for hardware devices. Lastly, they discuss the differences between two of the main programming software for VR.

A generic model was created by McKenney and Reeves [28] for conducting design research in education, which is depicted in figure 2.3. This model was also the basic framework for designing mobile VR learning environments by Cochrane *et al.* [29]. Comparing this model to the model of Vergara *et al.* [27] shows similarities but the model by McKenney and Reeves is more flexible and iterative in all stages. Another difference is that during each reflection moment it is tried to gather a new theoretical understanding about the topic. Next to the model, McKenney and Reeves [28] advise reviewing the exploration and analysis phase with an expert.

A third model was created by Mader and Eggink [30] for the study Creative Technology, which can be seen in figure 2.4. This model is an iterative process with multiple stages. The insights gathered in later stages can be used as new input for another iteration of an earlier stage. This way of flexibility in moving

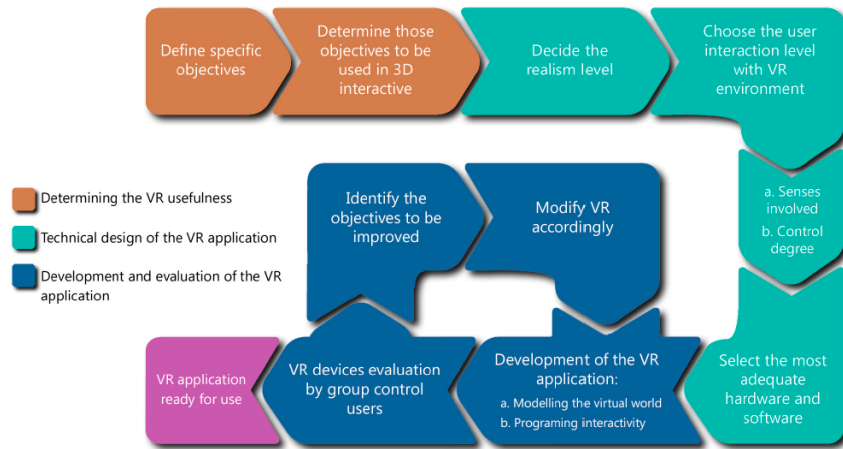


Figure 2.2: General flowchart for designing a VR application [27].

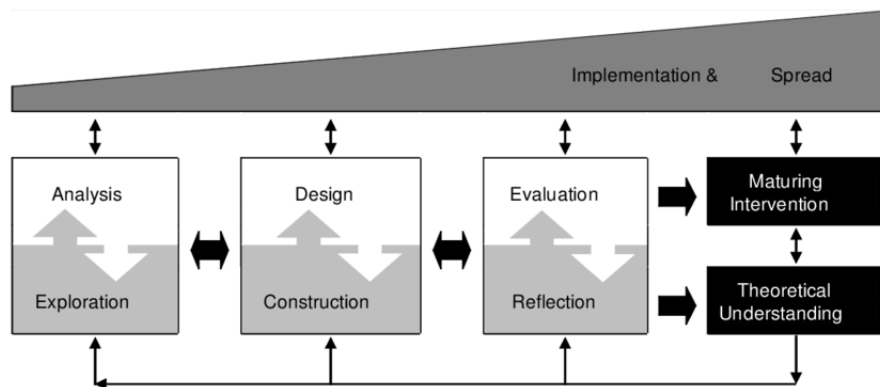


Figure 2.3: Model for conducting design research in education [28].

between stages is similar to the model of McKenney and Reeves [28]. The model is aimed at getting increasingly closer to the requirements by gathering new insights and feedback.

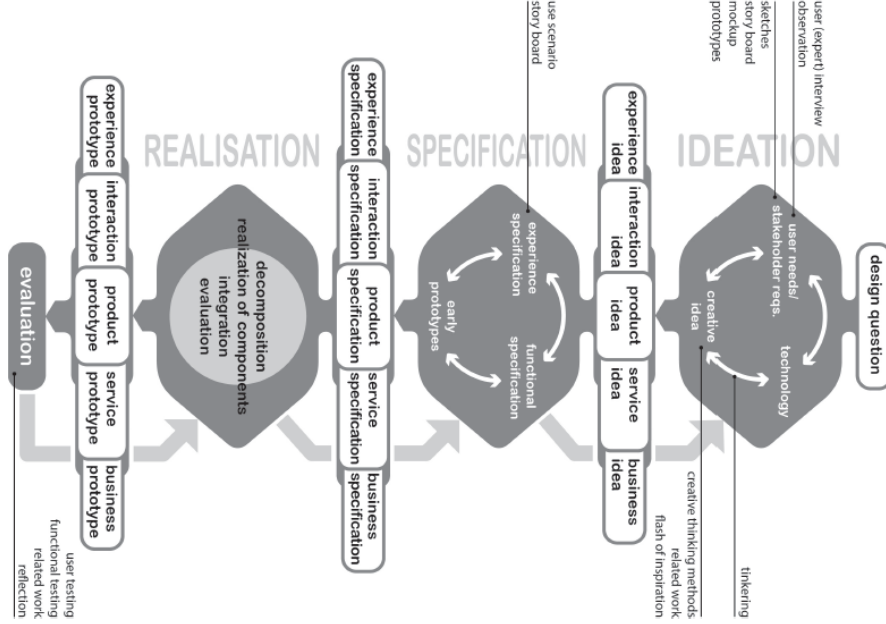


Figure 2.4: Creative technology design process [30].

2.4 User Interaction in Virtual Reality

Interacting with objects in VR is important while transporting items. The more advanced VR headsets have two common ways of having input in the virtual environment (VE). The first is using one or both of the controllers that are often provided with the headset. The second common way of interaction is using hand gestures. Hand gestures are most suitable for human-computer interaction [31]. For object manipulation of large sizes, participants of a study by Kang *et al.* [32] showed the preferred behaviour of using both hands. Dual-hand interaction is only advised when the task does not require precise movements. Physical aspects in VR are difficult to design for an immersive experience [26]. A VR user that lifts an item in VR does not have direct feedback on their body as in real life. Since it is all virtual there is no force working on the user's body. Without this force, it is hard to predict the heaviness of an item. To recreate a sense of heaviness, Weser and Proffitt created a method where the maximum speed of an object was determined by its mass [33]. Similar research was performed by Bäckström [34]. Her interaction method did show potential, the results however

did not show a clear confirmation of her theory. The study of van Polanen *et al.* [35] tried to accomplish the same effect. Their method was by visual delaying the objects. Both studies create the illusion of a heavy object by reducing the speed of the actual movement.

While interacting in VEs, the user can sometimes lose their sense of immersion. One reason for this is that some VR headsets are wired. Yoo *et al.* [36] found that wired headsets can negatively influence the safety and comfort of users. Another reason for losing the sense of immersion is that users in some cases develop cybersickness [37]. This is caused by multiple factors and also dependent on the VR system and the user [38]. Because this cybersickness is so dependent on multiple factors there isn't one solution. Arvilab produced a general list of methods to reduce cybersickness [39]. This list consists out of seven effective and three less effective ways of reducing cybersickness. Not all methods of the list are applicable to the goal of this project, but five methods could be. The first method is to reducevection, the illusion of self-motion when actual physical movement is absent. This can be achieved by reducing the speed and acceleration of the player's motion. The second is reducing the frames per second (FPS) of the camera to 10 to 15 FPS when the world is rendered at 90 FPS. The third method is eliminating the change of visual patterns. Reducing the complexity of textures and removing patterns helps to eliminate the change of visual patterns. In addition to this, the environment can be designed in low poly-style graphics. The fourth is creating a small field of view. The fifth method is to use 80% of dark colours and 20% of bright colours.

2.5 User Interfaces for Virtual Reality

Most people are not used to 3D environments and their user interfaces (UI). For that reason, the interfaces should be clear and easily understandable for all users. The creation of this can be hard since there is a lack of design guidelines and example projects according to Ashtari *et al.* [26]. Alger came up with different zones for displaying information in VE [40]. The different zones show where meaningful information in head-mounted displays is best shown with ergonomics in mind. One of the most important zones is displayed in figure 2.5. Displaying within the comfortable content zone is one of the seven guidelines provided by Fröjdman [41]. All these guidelines are aimed at providing a clear graphical user interface (GUI). When the user is interacting with an interface the position is important. Bernatchez and Robert found that the position of an interface is best linked to the user's body [42]. The two best frames for an interface are frames that are in line with the body of the user or are following the user's head. Next to this connection, their study found that menus are quickly scaled wrongly according to the user. Their advice is to let users manually adjust the position of the interface. According to Weiß *et al.* [43], there are three kinds of interfaces: 2D-, 3D- and speech interfaces. Screenshots of their used interfaces can be seen in figure 2.6. Their study suggests using the speech interface when a lot of text needs to be entered. 3D interfaces are recommended

when immersion and fun are important. A 2D interface is advised when a lot of objects need to be selected fast and accurately.

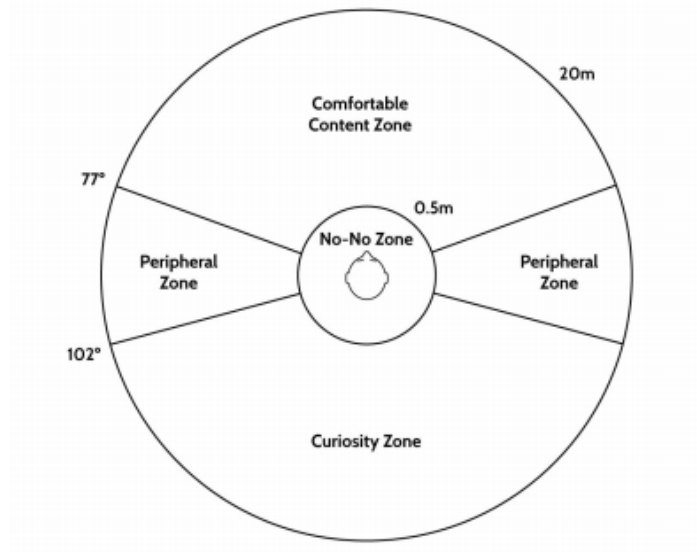


Figure 2.5: Placement meaningful information in VE [40].

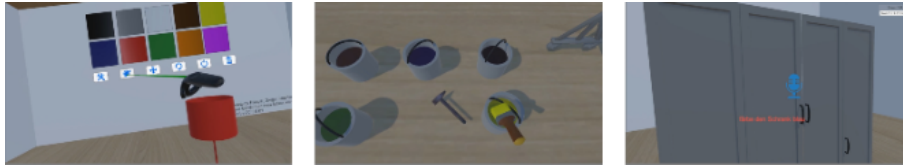


Figure 2.6: Screenshots of three kinds of user interfaces [43].

The study by Kharoub *et al.* [44] goes into greater detail on 2D interfaces. Their study researched three ways of interacting with this kind of interface, namely controller-, gesture- and point-and-click-based. The point-and-click system had the fastest completion time and the highest user experience, with more than 80% of the users expressing their satisfaction with this interaction.

To evaluate the usability of an interface in VR Sutcliffe and Kaur [45] created generic design properties that are important in a walk-through method. This method is not perfect but is able to filter some of the most important flaws out of the design. In addition to this, a study by Livatino and Hochleitner [46] provides guidelines for testing VR applications. These guidelines should assist in conducting pilot and formal studies. Lastly, the SUS test created by J. Brooke [47] can be used for testing the usability of the application.

2.6 Conclusion

The goal of this research was to find out the suitability of virtual reality, how to effectively learn in VR and how to effectively develop an application with good user interaction. Using a review of relevant sources it was determined that three studies do support the usage of VR. It can be argued if the costs outperform the expected learning outcome. The remaining rules for not using VR do not apply to the goals of this research. For these reasons, it can be concluded the reasons to use VR outperform the reasons to not use VR and thus is VR validated to be used in this research.

With studies supporting the usage of Virtual Reality, it was checked how users could most efficiently learn in VR. The best way to learn in VR was the learning by doing method in a goal-based scenario. To keep the users engaged during the learning process elements of gamification can be applied. With adding gamification, the possible negative effects should be tried to be avoided.

Next, it was checked how a VR application could be developed best. The most complete model that has been found for creating VR applications is created by Vergara *et al.* [27]. In addition to this model elements of the model of McKenney and Reeves [28] will be implemented, such as extra iteration possibilities and an expert review during the analysis and exploration phase. This new combined model should guide the development of the application.

The user interaction is vital in this application. To create realistic behaviour between the user and the VE, hand gestures are most suitable. Different studies showed multiple guidelines for using gestures in VR. To create the illusion of heaviness in VR multiple studies supported the method of reducing the speed of items. To keep the users immersed the use of wireless headsets is advised and the different methods of preventing cybersickness should be applied in the development of the application.

To create a good UI many aspects are important. Different guidelines have been found on the placement of UI. The placement is best when linked with the user's body or head. Next to that, there are different zones that create good placement for important information. The best way of interaction with an interface is dependent on its goal. Lastly, two studies have been found to evaluate the usability of a VR application.

Conclusively, the use of VR for this research has been validated. Different ways of effectively learning have been found that can be applied to instruct and engage the user. Next to that two models have been presented that can guide the development process of the application. Lastly, different guidelines were shown that should provide good user interaction for the user with the VE and the UI.

Chapter 3

Method

This chapter describes the method that will guide this research will be explained.

3.1 Research Method

The Creative Technology Design Method [30] will be used as a basis during this research. This model will be extended with added elements of the flowchart for designing a VR application [27] and the model of conducting design research in education [28]. This combined model will guide this research and is depicted in figure 3.1. This model consists of four stages.

3.1.1 Ideation

Ideation is the first step of the model and is aimed at creating ideas. The process starts with a design question. In the first spiral, different techniques for creating ideas can be applied. These techniques create multiple ideas and can be evaluated with stakeholders. This process can be iterated on until a project idea is created that seems feasible. This idea then continues to the specification stage.

3.1.2 Specification

With the project idea in mind, four steps will be answered that specify the requirements for VR technology. After the VR specifications, the general specifications for the project are determined with another iterative process. In this process, multiple prototypes will be created and evaluated. The evaluation of these prototypes should create the project specifications. These specifications will be used in the realisation stage.

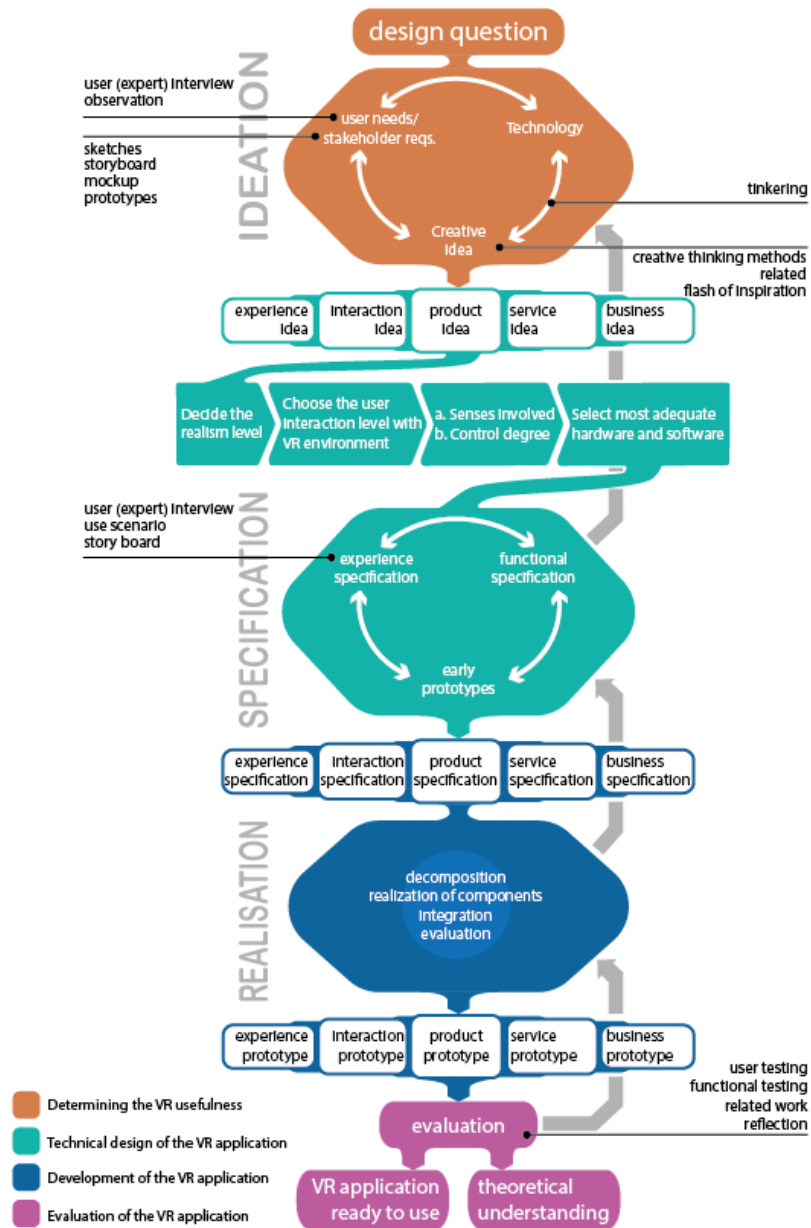


Figure 3.1: Used model for this research.

3.1.3 Realisation

The realisation is the third step of the model and tries to create a final prototype that fulfils the project specifications. Some of the early prototypes can be merged or partly reused for the final version. The final version will be evaluated on the earlier determined specifications. This project prototype will then continue to the evaluation step.

3.1.4 Evaluation

Evaluation is the last step of this model. In this model, the created prototype will be user tested and reflected on. If necessary, the needed changes will be made to improve the prototype by revisiting previous steps in this model. If the prototype meets all the requirements it's ready to be used. Next to the final product, a theoretical understanding will be created about the research.

Chapter 4

Ideation and Requirement Capture

This chapter describes multiple ideation techniques. These techniques lead to first concepts that were fine-tuned until initial concepts were created.

4.1 Co-design

During the development of the application, multiple prototypes will be created. These prototypes will be evaluated with various people from Innovatiehub Salland. This way multiple visions can be shared on the design, which gives a better view of the wanted application.

4.1.1 Stakeholder Analysis

Multiple groups have an interest in this project with different interests. For this reason, all the needs per group were gathered with the use of a stakeholder analysis. A stakeholder analysis was performed to find the requirements per stakeholder. Innovatiehub Salland as the host of this project had already performed a stakeholder analysis. During the co-design session, this analysis was worked out and shows the three stakeholders and their interests in this project.

Innovatiehub Salland

Before building apartments, Innovatiehub Salland wants to know if all items fit inside the building with the existing infrastructure. When this is confirmed the project can continue, when the building does not meet this requirement the building will be rejected.

Construction Workers

This stakeholder needs to build the apartments. Before building, they need to transport the items inside the vacant building. Some items may be difficult to transport. For this reason, they need to be able to practise this process before the actual moving. This way they can become familiar with the new environment and difficult areas of it. During the actual moving day, they require less time since they are already familiar with the location.

Municipality

Before the “Doos in Doos” project can start, the municipality must approve a building for the project. To convince the municipality they need to see that all the items can fit within the building assigned to the project. By letting them experience the fitting process themselves they can be convinced more easily. Next to the fitting process, the municipality wants to see what the building will look like with all apartments placed inside. Since a modular building process is used multiple configurations are possible and these all need to be able to be visualized.

4.1.2 Brainstorm

After the co-design session, multiple other brainstorm sessions were performed. A mind map, depicted in Appendix A, combines all these sessions in one visualization. The mind map shows different design aspects that potentially could be used for the creation of the application. The mind map is divided into two main sections, a section for checking if items fit inside the building and a section for training. The first section divides the possibilities in manually checking and in checking with artificial intelligence. The second section creates different modes and interaction possibilities. The design aspects that are not placed inside the two sections provide some general design considerations. In addition to some design aspects there are examples of VR applications linked that were created with these design aspects.

4.2 Initial Conceptualisation

Several ideas were generated in the brainstorming sessions. The focus of the initial concepts was on checking if the items can be brought inside a building since this is the main goal of this research. Two different modes were prototyped in the next stage, the 'Fit it' and 'Do it mode'. These modes were compared to each other in Chapter 5 to see what has the most potential.

The 'Do it mode' was made in a first-person view and lets the user experience it like they are transporting the items themselves. This mode could potentially be expanded to a two-person mode to recreate a more realistic scenario. Next to checking if the items fit, the 'Do it mode' could also be used to train construction workers. This mode can be controlled with the use of the controllers of the VR

headset or by walking in real life. The 'Fit it mode' was made in a miniature environment and will be controlled with two different inputs. The items that need to be transported can be controlled in two ways. One way is grabbing the objects with your hands using hand tracking. The second option is to control the object with a controller recreating the hand movement just as in the 'Do it mode'.

4.2.1 Storyboard

To visualise the initial concept a storyboard was created and is depicted in figure 4.1.

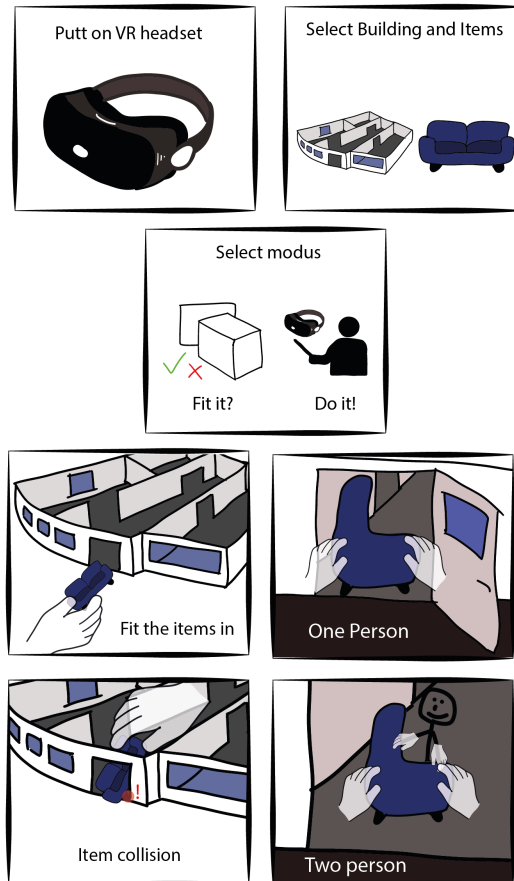


Figure 4.1: Storyboard about the initial concepts.

4.3 User Scenario

To create more insight into the two modes, a user scenario was created for both modes to give more context. This extra context can help shape the needs for the first prototypes. Both user scenarios show how a potential user could use the application.

4.3.1 Fit it Mode

Innovatiehub Salland wants to know if a vacant building is applicable for the 'Doos in Doos' project. Therefore they can let an employee use the 'Fit it Mode'. This user scenario is about this employee using this mode in the application. In this user scenario, the Innovatiehub created a point cloud model of the potential building before the employee starts the application.

1. The employee adds the point cloud model to the application and configures standard settings to the new environment.
2. The employee puts on a VR headset and selects the 'Fit it mode' after opening the application.
3. In this mode, the employee selects the items that need to be brought inside and the newly added building.
4. The last part of the setup is to select the begin and endpoint. The begin point is the place where the truck would be to get the items from and the endpoint is the location where the items need to be transported to.
5. The first item appears at the start point.
6. The employee zooms in to this point of the environment by grabbing the environment with two hands and pull them apart to create the zoom in motion.
7. He picks up the item with his fingertips with the usage of hand tracking and brings the item to the entrance of the building. The employee creates another angle on the environment to clearly see the entrance by rotating the environment.
8. With the ray beam coming out of the virtual hands of the employee, he lifts the items through the entrance of the building.
9. When the item hits the doorpost the item becomes red meaning he hit the environment.
10. The employee tries it again by rotating the item and lifts the item through the doorpost.
11. The item is brought to the endpoint and the moment it hits the point beam the item disappears and the next item appears at the start point.
12. The employee repeats this process for the rest of the items.
13. The moment the last item hits the endpoint, a pop-up message appears stating that this building is applicable for the 'Doos in Doos' Project.
14. This pop-up message is clicked away and the employee closes the application through the menu.
15. The employee turns the headset off and tells the boss of the project this building is applicable for the project.

4.3.2 Do it Mode

The 'Do it mode' can also check if items can be brought inside the building but can also be used to train employees. This user scenario is about the training simulation of a construction worker a few days before the actual moving day of the items.

1. The construction worker puts on a VR headset in a big open space.
2. A virtual wall is created so the headset knows the area it can safely use when the user walks around.
3. The application is started and the 'Do it mode' is selected in the menu.
4. The first item appears next to the truck and the construction worker walks towards it.
5. When standing next to the item the construction worker grabs the item with the virtual hands.
6. The construction worker walks towards the entrance of the building.
7. By turning the virtual hands, the construction worker can rotate the item so it fits through the entrance.
8. The items can be moved around with one or two hands by the construction worker.
9. The construction worker releases his grip on the item by accident and the item falls naturally to the ground.
10. This item is brought to the endpoint where the item disappears and the next item appears next to the truck.
11. The construction worker repeats this process with multiple items until the last item appears.
12. When the construction worker tries to bring in the last item, he is not sure how to get it through the entrance.
13. With the push of a button, a simulation is shown in front of the construction workers eyes.
14. The construction worker replicates this shown movement and brings the item inside the entrance.
15. The moment the item touches the endpoint a pop-up text is shown to the construction worker that states that his training is completed.
16. This pop-up message is clicked away and the construction worker closes the application through the menu.
17. The employee turns the headset off and stores it back to where it belongs.

Chapter 5

Specification

This chapter describes the constraints regarding the technical design. After these constraints, the first prototypes are created and evaluated on with a co-design session. Lastly, the specifications for the final applications are created and ranked with the MoSCoW method.

5.1 Constraints on Technical Design

Before the specification iteration cycle can be entered some technical design choices need to be made. These choices determine how the application needs to be developed. The first step in this process is to determine the level of realism of the application. The construction workers want to be familiar with the building before entering it in real life. For this reason, the building should fully match the virtual environment with the actual one. The main importance for this is how walls are placed not what kind of texture the wall has. So the realism level is up to the shapes of the building not to textures.

The second step in the technical design choices is to decide the user interaction level. The user needs to be able to interact with the environment and the items. For this, the user needs a VR headset to visualise the environment and some control mechanism to interact. The VR headset must provide hand tracking since this is the most suitable way of human computer interaction [31]. VR headsets provide two possibilities for degrees of freedom. There are headsets with three degrees of freedom; with these it is only possible to watch in the virtual environment. For this application, a headset with six degrees of freedom is required to also be able to move around.

5.2 Prototypes

The prototypes are developed quickly with a focus on functionality. The basic interaction of manipulating the environment and items are applied to the pro-

totypes. This way the best interaction can be tested for a final prototype. Two prototypes have been created based on the initial concepts.

5.2.1 Fit it Mode

This mode starts with a God view, meaning the user looks down at the environment in miniature. Interaction is possible by using controllers or hand tracking. Hand tracking tracks the real hand of the user and creates a virtual one with the help of multiple cameras. The user is able to interact with this environment with his virtual hands and with beams that come out of these hands. The user can pick up the items with the tip of the fingers or by pointing the beam on the items and then recreate a grabbing motion. The environment can be scaled, rotated and moved around with hand gestures. The items in this environment can be picked up and also be rotated and moved around. The items collide with the ground and walls of the building, the hands do not. This mode is created with the Mixed Reality Toolkit developed by Microsoft [48].

5.2.2 Do it Mode

This mode was created in a first-person view, meaning that it seems like the user is present. The locomotion system, a system to move in a virtual environment, was based on moving in real life or with a joystick. The user can pick up the items at any position of the item with their virtual hands. The items are controllable with one and two hands just as in real life. The hands and items collide with the environment. This mode is created with the XR Interaction Toolkit developed by Unity [49].

The user is able to walk around in the virtual world by using a joystick on the controller and by walking in the real world. When it is not possible to move around in the real world because of limited space it is possible to use the joystick. When this method of control is used, it is possible that forms of cybersickness occur because in the virtual world it looks like the user is moving while in real life the user is not. For this reason, some of the methods to reduce cybersickness by Arvilab [39] are applied to this mode. The materials of the items and the environment are made with a clean and one coloured texture and a low poly-style design. The speed and acceleration of the player are reduced heavily and lastly, the field of view is made smaller.

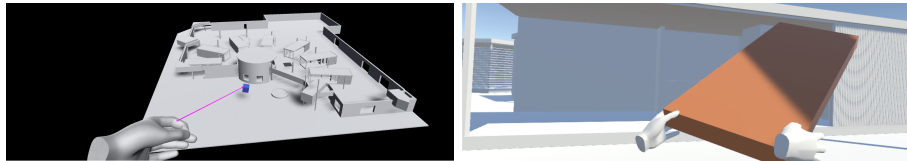


Figure 5.1: Left: Fit it mode. Right: Do it mode.

5.3 Co-Design

5.3.1 User testing

During the co-design session, the host of the project and a VR expert were present to test the prototypes. This limited group was selected to adhere to the COVID-19 guidelines that were present during this co-design session. Both users tried to bring in all the items in both prototypes with the different controls. The conclusion about the models is presented below.

Fit it Mode

Overall the 'Fit it mode' worked smoothly. Only with hand tracking, it would sometimes drop the item because the hand tracking algorithm is having trouble when fingers are touching each other.

Do it Mode

The basics of the application worked, the user was able to walk around the environment with both methods and pick up items. Sometimes when items were picked up some glitches appeared. The items could sometimes start spinning around the wrist or would stick to the hand and not leave the hand when the item was released. These glitches were not the biggest problem because when some users walked around with the joystick forms of cybersickness started to occur. Even though some methods advised by Arvilab [39] were applied these forms of cybersickness could not be prevented during the movement with a joystick.

Conclusion

After testing the prototypes, both users thought the 'Fit it mode' worked better. Both users suggested that the 'Do it mode' still has potential but that the glitches and cybersickness should be removed. Each user preferred a different control mechanism. For this reason, both control options will be applied in the final application and can easily be switched between.

Overall the decision was made to first focus on the 'Fit it' prototype. This was for the main reason that the 'Do it' prototype created forms of cybersickness and still had multiple glitches. The functionality of the 'Do it' function will be added to the end product by zooming in to a first-person view or by creating an improved mode. Next to this, an additional mode to place apartments inside the building will be added to the application. This mode places whole apartments inside the building instead of single items that are part of the apartments. This new mode should help convince the municipality to accept the "Doos in Doos" by providing them with the option to design the building themselves quickly.

5.3.2 Requirements

During the co-design session, the requirements for the end product were specified. These requirements were sorted with the MoSCoW technique to create priorities between them. The MoSCoW method divides the requirements into the categories 'must have', 'should have', 'could have' and 'would have'.

Must have

- The application needs to work on the Oculus Quest 2.
- Zoom, pan and move the environment with items in it.
- Item interaction between hand/controller and the items.
- Have a start and endpoint for the item transport. (The start point is where items are dropped off by the truck and the endpoint is the location where the items need to be transported to.)

Should have

- Collision detection between items and environment.
- Two-handed interaction between the user and the items.
- Realistic behaviour on items when they are being moved around or dropped in the virtual environment. (Creating a sense of heaviness on items)
- Show if a building is applicable for the project when all items are brought inside the vacant building.
- Being able to add apartments to an empty building.
- Create a menu to switch between modes and close the application.

Could have

- Create guidance for construction workers that are stuck.
- Add an import function for new environments/buildings.
- Create a tutorial for the application.
- Create a progress bar to see how many items are already brought inside. (Could be visualized in the form of an apartment being build up.)

Would have

- Option to add a point cloud model as a virtual environment.
- AI behaviour to see if an object fits through certain parts of a building.
- A two-person mode for recreating a realistic scenario in the 'Do it mode'.

Chapter 6

Realisation

This chapter describes the final version of the prototype. This version is built further upon the 'Fit it' prototype discussed in Chapter 5. The most important functions of the application are described in the following subtitles.

6.1 Technical Design choices

Based on the constraints on the technical design the most adequate hardware and software was selected. For the hardware, the Oculus Quest 2 were chosen. This VR headset provides hand tracking and has six degrees of freedom. The extra benefits of this VR headset are that it is already being used by Innovatiehub Salland and it provides stand-alone running of applications and wireless connection to a PC through a private network.

Two main software programs are being used for VR development, namely Unreal Engine and Unity. Both are good options but for Unity, there is more support on the internet and a community present that could help for quick development. This is the reason Unity is chosen for the software.

6.2 Controls

The application is created with the Mixed Reality Toolkit from Microsoft [48]. This toolkit has been chosen for this project since it is able to assist with the controls in VR. The toolkit assists with hand tracking for the Oculus Quest 2 headset. Next to that, it lets the user easily switch between controllers and hand tracking. Another reason this toolkit is selected over others is that it has attributes for scaling items which contributes to the miniature version.

The application can be controlled in multiple ways. As described in chapter five, the user can use hand tracking and controllers to interact in the application. Both control mechanisms are available since users often have a personal preference. With either these controls, the interaction with items can vary between interacting with the fingertips or the pointer beam, these options are depicted

in figure 6.1. The fingertip interaction works similar to when the user is holding in its hand in real life. The interaction with the pointer beam works like a flexible stick. When moving the user's hand in a direction the item will follow in a smooth elastic manner. When the item needs to get rotated, the wrist of the user can rotate and the item will follow this rotation.

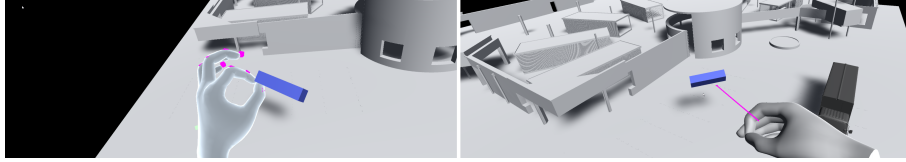


Figure 6.1: Left: Fingertip interaction. Right: Pointer beam interaction.

6.3 Modes

The final version of the prototype has three modes: the 'Fit it', 'Place it' and 'Do it mode'.

6.3.1 Fit it Mode

The 'Fit it mode' is created for two main purposes. The first one is to see if the items of the apartments fit through the existing infrastructure of a vacant building. The second is to let construction workers experience this fitting process in a virtual environment before the actual moving. During the fitting process, the user can zoom in to difficult points and look from different perspectives by turning the environment around.

Item Sequence

To test if a building is applicable all items need to be brought inside to the final location where the apartments need to be built, this is the endpoint. The starting point is outside where the truck will park, this is the starting point. When the application is started, the first item that needs to be brought inside will appear. When this item is brought to the endpoint the next item will appear at the start location. This process will continue until all items are brought inside. When this is the case, the building is applicable and this will be shown by a message that is depicted in figure 6.2.

All the 3D models of the items that need to be fitted inside the building can be stored in one folder. The code for the item sequence automatically puts all these items in a list in such a way that they will spawn one after the other. The code for the item sequence is depicted in Appendix D.



Figure 6.2: Message showing the building is applicable for the Doos in Doos project.

Object Collision

When items are being moved in the application it can sometimes be hard to see if the item is colliding with a wall or anything else. To make this visually clear when this happens the colour of the object will change to red. The moment the item is not colliding with anything it gets its original colour back. An example of an object collision can be seen in figure 6.3. The objects collide with everything except for the hands and the ground. The code for object collision is shown in Appendix D.

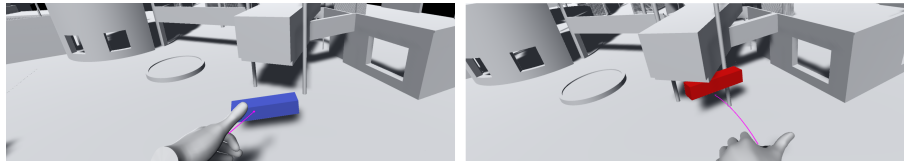


Figure 6.3: Left: No collision. Right: Collision with a pole.

6.3.2 Place it Mode

The 'Place it mode' is designed to place apartments in the vacant building. With this feature, it is possible to experiment with the setup of the apartments and visualize different outcomes of the project. This mode should help convince the municipality to accept the "Doos in Doos" project for specific buildings. A visualization of this mode is depicted in figure 6.4.

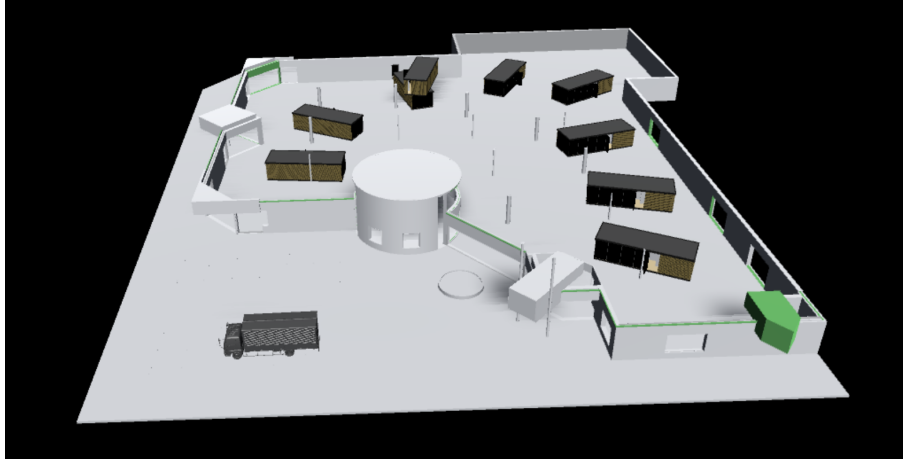


Figure 6.4: Visualization of a possible apartment layout.

Apartment Spawner

To be able to place apartments they need to appear in the virtual environment. The user is able to select the truck ¹ in the environment with their fingertips or laser pointer. When the truck is selected an apartment will appear next to the truck. This apartment can then be placed inside the vacant building. The code for the spawning of apartments can be found in Appendix D.

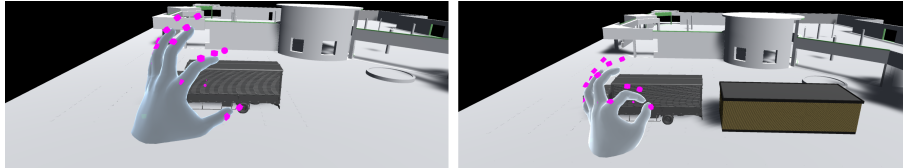


Figure 6.5: Left: No selection. Right: Select with pinch gesture.

6.4 Do it Mode

While using the Mixed Reality Toolkit it was noticed that fewer glitches appeared. For this reason, the 'Do it mode' was recreated in this new toolkit. This switch deleted most of the glitches. Also, the locomotion system was switched to teleportation from moving with the joystick when limited space is available.

¹Isuzu truck created by Nsfr750; <https://www.cgtrader.com/free-3d-models/vehicle/truck/isuzu-truck-177655bfef3abf46e631f9ff972667d1>

When the building is applicable for the “Doos in Doos” project, the construction workers can be trained. The 'Do it Mode' is designed to create an identical experience to the actual moving day. This way the construction workers can already experience the moving process before the actual moving day. In this mode, one and two-handed interaction is possible with all the items. Just as the 'Fit it mode', this mode also has an item sequence and object collision build in. At the end of the sequence, the message 'Training Completed' will pop up as depicted in figure 6.6.



Figure 6.6: A message pops up when the sequence is completed.

This mode was designed to be used in an open space because the main movement of this application is to walk in the real environment. Before using the application the walls of this open space need to be defined so virtual walls can be placed that prevents the user from walking into an actual wall or other obstacles. When there is limited space the teleportation method can be used.

6.4.1 Teleportation

The previous method of moving around in the 'Do it mode' created cybersickness. Another suggested method of moving around in VR by Arvilab [39] to reduce cybersickness is to use teleportation. This method lets the user determine where it moves to with a ray beam, as depicted in figure 6.7. This ray beam can be directed by the index finger of the user while using hand tracking or by pushing the joystick forward while using the controllers. When activated, the user will teleport to the new location that was marked by the teleportation ray.

6.5 Graphical User Interface

To be able to switch between the different modes a menu is created, which is depicted in figure 6.8. This menu gives the opportunity to switch between the three different modes and close the application. As the study of Weiß *et al.* [43] suggested the best interface for fast and accurate selection is a 2D interface. For this reason, a 2D interface was designed but with 3D buttons too also suggest the buttons can be clicked while using hand tracking controls. The position of the

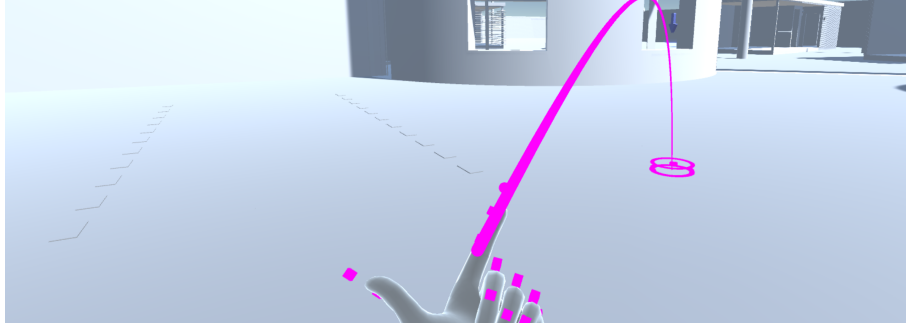


Figure 6.7: Teleportation ray to move around.

menu is placed in the comfortable content zone created by [40]. The suggestion by Bernatchez and Robert [42] to let the menu follow the head of the user is also applied but can be switched on or off by the user by pressing the toggle button on the top right. This option is given to the user because the same study by Bernatchez and Robert [42] also suggested letting the user position and scale the menu themselves. The menu can be scaled and positioned by using two hands and grabbing the outside of the menu and scale it with two hands. The menu will follow the head of the user again when the toggle button is pressed with the same scale the user scaled the menu too. The menu can be activated and deactivated by pressing the menu button on the controller or while using hand tracking looking into the left palm while pressing the index finger against the thumb. The code for activating the menu and using the buttons can both be found in Appendix D.



Figure 6.8: Visualisation of the menu

6.6 User Testing

During the user testing, the application was tested with the task list depicted in 9.2.3. During these tasks, the participants were asked to think out loud, so their immediate thoughts could be collected on the application. After the task list, they were asked about the flaws of the application and things that could be added or improved.

6.6.1 First Iteration

The participants that took part in the user testing were other students also graduating at the client of this project.

Feedback

The original menu was quite hard to read according to the participants. The text inside the button was sometimes hard to read, therefore some participants suggested moving the text outside the button. Another participant suggested creating the text in 3D.

When performing tasks in the 'Do it mode' the lighting was different from the other modes. This mode gave some participants the feeling they were in some horror scene. Therefore they mentioned they would prefer other lighting that would create a brighter scene. An example of the lighting in the scene is depicted in figure 6.9.

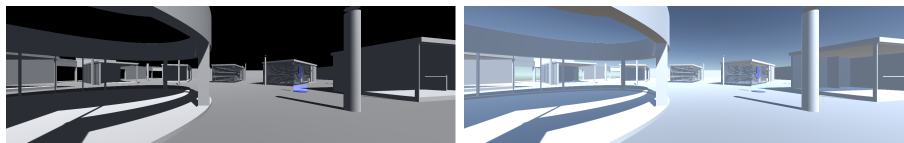


Figure 6.9: Left: Old lighting in the 'Do it mode'. Right: Updated lighting in the 'Do it mode'.

Changes

To create a better readable menu the text of the buttons is moved outside the button they were originally in. The text is now created in 3D and is placed under the button but still in the front so the text does not disappear under the button when looking from above. The text change of the menu is depicted in figure 6.10.

To create more overall lighting in the scene a skybox was added just as in the other scenes. The result of this is depicted in figure 6.9.



Figure 6.10: Improved menu with the reflection of the light.

6.6.2 Second Iteration

For the second user testing, the participants were students that are part of the household of the researcher.

Feedback

The updated menu still was not clear to some participants. A participant suggested adding icons to make a better division between the different modes since they now all looked similar with the squares. Next to that at a certain point, the menu was not readable because of a sun flair on the buttons, as depicted in figure 6.10.

While participants were performing the task list in the 'Do it mode' two participants mentioned they missed the roof on the building. When adding this, this would create a more realistic building.

Changes

To prevent the lighting reflect on the buttons, the material of the buttons was changed. To create a better distinction between the buttons an icon was placed inside the button instead of the white square. The 'Fit it mode' was given a checkmark icon, the 'Place it mode' a house icon, the 'Do it mode' two hands as an icon, the exit button an exit icon and the focus button was given an eye icon. The final menu is depicted in figure 6.11.

To create a more realistic building in the 'Do it mode' a roof was placed on this building. Because of the roof, the lighting needed to be adjusted again. To create natural lighting inside a building, a directional light from the top was placed inside the building. The final result of the 'Do it mode' can be seen in figure 6.12

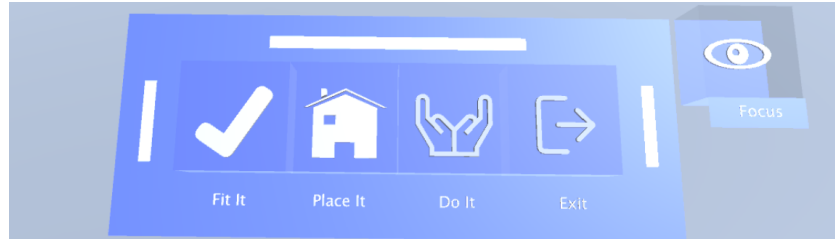


Figure 6.11: Final menu with icons and no reflection.

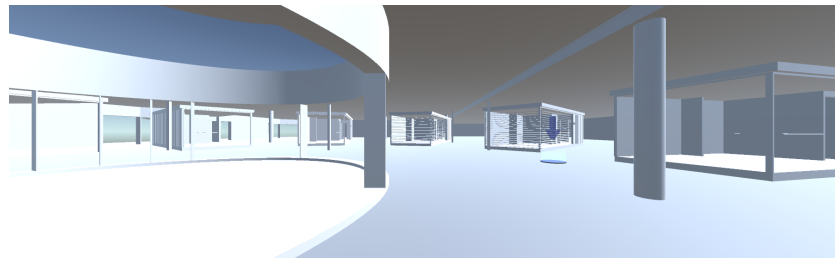


Figure 6.12: 'Do it mode' with directional light from the top.

Chapter 7

Evaluation

This chapter describes the evaluation of the application. The application will be evaluated in different ways. The first evaluation is on the implementation of the set requirements. Secondly, the feedback from the user testing is evaluated.

7.1 Requirements vs Implementation

In chapter 5.3.2, the requirements were listed for the application. During the development process, these requirements were tried to be applied in the software. Not all requirements have been implemented in the final application due to a limited time span.

When looking at the requirements with the MoSCoW method, the most important requirements have been implemented. All 'must have' and 'should have' requirements have been implemented in the application. The requirement for realistic behaviour in the environment is partly achieved and could be extended. In the application is a realistic behaviour of walking and handling items with one and two hands that matches real-life. In addition to this, the application could be extended with the feeling of heaviness. This can be created by reducing the speed of the actual movement as multiple studies have suggested [33] [34] [35]. Also when an item is dropped the fall of this item is not fully natural. When an object is thrown the object directly goes to the ground instead of following the throwing motion. Lastly, haptic feedback in vibrations could be added to when an object collision occurs or when an item is brought to the endpoint. Adding these realistic behaviours could improve the immersion of the application.

The 'could have' and 'would have' requirements from the MoSCoW method were not implemented but the application could still benefit from those requirements. Creating a tutorial inside the application could create the benefit of needing no explanation before using the application for new users. When the application is used a lot and often new environments need to be added this process needs to be improved. In addition to this, also point cloud models of a building need to be able to be added for a first quick check if items fit. Another

aspect that is currently missing is the 'Authoring by doing' method. Without this, the construction workers that are stuck can not get help. This could be realised by creating artificial intelligence behaviour for the items so it can be checked if they fit and how. This how part could be contributing to the authoring by doing method. Lastly, a progress bar for the user to see how many items are already brought inside could also improve the experience by showing the user the progress of the task.

In conclusion, the most important requirements were implemented in the application to fulfil the primary tasks. The application could be improved by implementing the 'could have' and 'would have' requirements.

7.2 Feedback from User Testing

This evaluation was mostly focused on non-numerical, subjective data. Because of the ongoing COVID-19 pandemic during this research, there was a limited amount of participants for reviewing the application. All participants were selected so no extra physical contact would be created because of this research. This resulted in a group existing of students from the University of Twente. The process of user testing can be found in Appendix B.

7.2.1 General Experience

All users enjoyed using the application, this was mostly due to the fact it was the first encounter with VR for almost all participants. Almost all users could see the added benefit for the employees of Innovatiehub Salland. Some remarks however were given about VR technology. Some users questioned how long the employees would want to use the application because after a while their eyes would start to hurt from using the VR headset.

7.2.2 Feedback on Application

After testing the application in VR the participants were asked to answer questions about the application. These questions can be found in Appendix B.

Interaction

Most participants preferred the interaction with controllers over the interaction with hand tracking. This was mainly for the reason that the accuracy of controllers was more accurate and does not show small glitches. Examples of these glitches were dropping items while carrying and gestures that need to be repeated multiple times before getting accepted by the software. These glitches exist because the hand tracking software is not perfectly developed yet. As some participants pointed out, the use of hand tracking is more intuitive than using controllers so they do see the added benefit. For now, they prefer the controllers for the main interaction. The fast switching between both types of controls was pointed out as a benefit by some users. Some participants praised that the main

interaction only exists out of one or two kinds of interactions, which for them made it easy to interact.

Modes

All modes were obvious to the participants and most of the participants could see the added benefit for all the modes. However, some participants pointed out that the result of the 'Fit it mode' could have been gathered with the 'Do it mode'. Because when a hard part comes to manoeuvre an item inside the user often almost zooms in to a first-person view, just like in the 'Do it mode'. Others did see the benefit of this mode because when an item was passed this hard part it could faster be moved to the next point than in the 'Do it mode' where the user needs to teleport a few times to achieve the same.

The object collision in the 'Fit it mode' was triggered too early in some cases. This is due to the fact that when the environment is zoomed out the rendering is less precise and thus the item thinks it is hitting something else while this is not the case. In attempts to solve this, the rendering became too heavy for the hardware and started to lag. This made it impossible to use the application.

The 'Place it mode' sometimes showed some small glitches. The apartments are restricted to only moving around their y-axis. When using hand tracking, the apartments sometimes do move around their y-axis after releasing it from the virtual hand. This sometimes causes the apartment to be placed at an angle.

Graphical User Interface

The menu was easy to navigate according to the participants. The movement of using a finger to push a button while using hand tracking was not directly clear. The function of the focus button was also not directly clear but after trial and error this became clear. Some participants found the scale and placement option unnecessary, while others found it a nice addition. This is in line with the finding of the study by Bernatchez and Robert [42], who advised to let the user adjust the menu when necessary. A small bug that was discovered is that after the focus button is pressed the menu can not be rotated around the x-axis, which sometimes resulted in a harder angle to read to the menu.

Learning

All participants knew how to bring in the items. This was an easy job since no specific pose was required to bring in the items. They did have an increased sense of how the building was structured and how they would need to walk during an actual moving day. The participants stated that especially the 'Do it mode' boosted this since they needed to walk the route from start to endpoint multiple times.

7.2.3 System Usability Scale

To test the usability of the application a SUS test was applied. The SUS test [47] was created to give a quick, uniform and reliable insight into the usability of a system. Ten Likert scale questions were asked to the participants about their satisfaction with the application. The participants were asked to answer these questions as if they were end-users because the test required in context answers. The questions and how to validate the questions and the score can be found in Appendix B.

In total, ten participants completed the SUS test. The results of this test are depicted in table 7.1 and ranged from 70,0 to 97,5 with an average of 83,75. The score indication for a SUS test is depicted in figure 7.1 [50]. When placing the average result of the SUS score into the score indication figure, it indicates that this application has acceptable usability and is placed between good and excellent in the adjective ratings.

Participant	Score
1	90,0
2	72,5
3	75,0
4	80,0
5	97,5
6	70,0
7	92,5
8	85,0
9	82,5
10	82,5
Average	83,75

Table 7.1: SUS results.

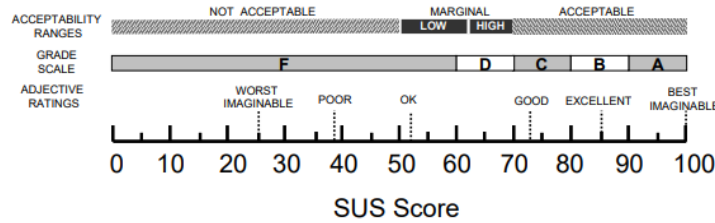


Figure 7.1: SUS score indication [50].

7.3 Feedback from the client

The application was evaluated with the client through the questions described in Appendix B.

Overall the client was pleased with the final application as it met his initial expectations. All the main functionalities of the application are working good and no changes need to be made. To further improve the application the additional requirement could be implemented. The client would like to merge the VR application created by Jelle Smith with the 'Fit it mode' of this application to give a better visual of the potential configurations. The client feels that the application has an added benefit for the company and wants to apply it on a project.

Chapter 8

Conclusion and Discussion

To conclude this project, the research questions will be answered. By answering the sub-questions, it will be possible to answer the main research question. After this conclusion, the project will be reflected on and lastly, recommendations for future work will be presented.

8.1 Conclusion

The main objective of this research was to create a VR application that is able to indicate if items can be transported into a vacant building. This objective was achieved by creating a VR application that could fulfil this and additional tasks with co-creation. This goal was achieved by answering a number of sub-questions.

How to effectively learn to transport items in VR?

According to literature, effective learning can be achieved by the method of learning by doing. The goal-based learning method, which is a form of learning by doing was applied to this application. This method resulted in a better understanding of the layout of the vacant building by participants of the user testing.

How to develop a VR application for transporting items?

The model depicted in figure 3.1 shows the method for developing a VR application for transporting items. In the technical design choices of the model, it was decided a VR headset with 6 degrees of freedom was needed with hand tracking for transporting items. Unity is the best software to use for quick development since it has more support available on the internet and a big community.

How to create user-friendly interaction in VR for transporting items?

To create user-friendly interaction the number of different interactions should be kept low. Most users prefer the interaction with controllers over hand tracking because of the higher accuracy. Some participants like hand tracking more because it is more intuitive. Fast switching between both kinds of controls

contributes to user-friendliness. For the graphical user interface, the guidelines provided from the literature research were applied. This resulted in an easy to use interface. Lastly, at all costs forms of cybersickness should be prevented for a user-friendly interaction. In this application, the applying of the teleporting system, simple graphics and low poly design did help prevent the occurrence of cybersickness.

What is the reaction of the potential end-user(s) on the created application?

The participants of the user tests and the client enjoyed using the application and did see the added benefit for Innovatiehub Salland to use this application. The client wants to further develop the application and apply it on a project. The SUS test performed during user testing suggests that the usability of the application is graded between good and excellent.

Finally, this leads to answering the main research question.

How to create a VR application for transporting items?

In this graduation project, a VR application has been created for checking if items of an apartment fit with the infrastructure of a vacant building based on the findings of the sub-questions. In addition to this, a mode is created for visualizing vacant buildings with apartments and a mode for training construction workers. This application is created with co-design and multiple iterations based on user testings.

8.2 Discussion

The participants of the user tests do not match with the potential end-users. Therefore the results of the SUS test can be questioned. The participants that took part in the user testing were all relatively young in comparison to the potential end-users. Knowing this, the presumption is taken that these participants of the user tests are more easily familiar with a new technology such as VR. This suggests that the score of the SUS test could be negatively influenced when tested with potential end-users. Therefore it is recommended to test the application on end-users.

The designs of the 'Doos in doos' apartments were not finished during the development of this application. For this reason, the application was not able to test with the real items that need to be brought inside. When these items are available, they should be added to the application and tested with to see if everything still works as designed.

The model that was used during the research, depicted in figure 3.1, has a small flaw and needs to be updated. The last of the four steps between the ideation and specification phase needs to be moved to the realisation phase. This is for the reason that the most adequate hardware and software can only be chosen when the specifications for the final product are known.

This application is created to be a tool and should never be fully trusted as the solution. When there somehow is a mistake in the 3D model of an item or the building this should not result in construction workers forcing the item through

a gap because it should fit. When construction workers believe it will not fit they should always trust their senses and stop transporting when necessary. If they would continue their transporting process they could potentially end up damaging the building and/or items.

8.3 Recommendations

Several recommendations can be made. Firstly, it was not possible due to COVID-19 to test with potential end-users of the application. As stated before, it is recommended to test on these end-users to get insights from this group. These insights can make sure the details of this application are tailored to this group. In addition to extra testing, a new environment needs to be added that is more challenging to bring in the parts of the apartments. With a more challenging environment, the users of the application can test if their sense of transporting items will improve.

A study about the learning effects of this application could be performed. The insights of such a study could share insights into the learning effect of VR on transporting items. When the results on the learning effects are positive it could spark more learning applications in VR.

Not all requirements were applied in this application, mainly because of the limited time span of this research. The remaining requirements could be added to the application to create a better experience while using the application and add functionality. Creating a tutorial inside the application could create the benefit of needing no explanation before using the application for new users. Fully completing the requirement for adding realistic behaviour on the items and adding a multiplayer option to the 'Do it mode' could give the construction workers a more realistic training experience. Another aspect that is currently missing is the 'Authoring by doing' method. Without this, the construction workers that are stuck can not get help. This could be realised by creating artificial intelligence behaviour for the items so it can be checked if they fit and how. This how part could be contributing to the authoring by doing method. Lastly, a progress bar for the user to see how many items are already brought inside could also improve the experience by showing the user the progress of the task.

To create a more engaging training experience, the use of gamification could be added. This way the construction workers could for example race for the fastest time and with the least collisions. This gamification should not negatively influence the learning experience and this influence should therefore also be tested before actually applying it into the application.

The application created by J. P. Smith, to customize the interior of 'Doos in Doos' apartments, could be integrated into this application. This could be in the form of a fourth mode, but it could probably have even more impact as an addition to the 'Place it mode'. This way the building can be customized with apartments and every apartment can be customized to the preferences of the residents.

Since the apartments are not fully developed, this application could not make use of the parts that it will consist out of. These 3D models need to be added to finalize the product. Lastly, the application needs to be used on real applications. This will be the real test and should show if the application can perform for what it was designed.

Bibliography

- [1] S. Kromhout *et al.*, “Stand van de woonruimteverdeling,” RIGO Research en Advies, Amsterdam, Tech. Rep., Mar. 2019.
- [2] D. Evers *et al.*, “Veerkracht op de Proef gesteld,” Planbureau voor de Leefomgeving, The Hague, Tech. Rep., Dec. 2020.
- [3] M. Jansen, “*Doos in doos*” *Introduction Interview*, Dec. 2021.
- [4] Int. Data Corporation, “Worldwide Spending on Augmented and Virtual Reality Forecast to Deliver Strong Growth Through 2024, According to a New IDC Spending Guide,” IDC, Framingham, Tech. Rep., Nov. 2020.
- [5] C. Pérez-sabater *et al.*, “Active Learning to improve long-term Knowledge Retention,” in *Proc. of the XII Simposio Int. de Comunicación Social*, Santiaio de Cuba, 2011, pp. 75–79.
- [6] A. Kishore, *Boeing: Productive VR Cuts Training Time by 75%*, Jun. 2017.
- [7] M. Nooriafshar *et al.*, “The Use of Virtual Reality in Education,” in *American Society of Business and Behavioral Sciences*, James Cook University - Centre for Business and Economic Research, 2004, pp. 1–12, ISBN: 0646437178.
- [8] W. Winn, “A Conceptual Basis for Educational Applications of Virtual Reality,” Washington, Aug. 1993, [Online]. Available: <http://www.hitl.washington.edu/projects/education/winn/winn-paper.html>.
- [9] I. Duncan *et al.*, “A taxonomy of virtual worlds usage in education,” *British Journal of Educational Technology*, vol. 43, no. 6, Nov. 2012, ISSN: 00071013. DOI: 10.1111/j.1467-8535.2011.01263.x.
- [10] V. S. Pantelidis, “Suggestions on When to Use and When Not to Use Virtual Reality in Education,” in *VR in the Schools*, 18th ed., vol. 2, 1996.
- [11] V. S. Pantelidis, “Reasons to Use Virtual Reality in Education and Training Courses and a Model to Determine When to Use Virtual Reality,” in *Themes in Sci. and Technology Education*, Special Issue, Klidarithmos Computer Books, 2009, pp. 59–70.

- [12] F. Mantovani *et al.*, “2 VR Learning: Potential and Challenges for the Use of 3D Environments in Education and Training,” in *Towards CyberPsychology: Mind, Cognitions and Society in the Internet Age*, Amsterdam: IOS Press, 2001, ch. 12.
- [13] M. Horne *et al.*, “The Role of Virtual Reality in Built Environment Education,” *Journal for Education in the Built Environment*, vol. 3, no. 1, pp. 5–24, 2008, ISSN: 1747-4205. DOI: 10.11120/jebe.2008.03010005.
- [14] T. Hilfert *et al.*, “Performance analysis of OFDM modulation on indoor broadband PLC channels,” *EURASIP J. Adv. Signal Process*, vol. 78, 2011. DOI: 10.1186/s40327-015-0031-5.
- [15] R. C. Schank *et al.*, “Learning by Doing,” in *Instructional Design Theories and Models: A New Paradigm of Instructional Theory*, Mahwah, NJ: Lawrence Erlbaum Associates, 1999, pp. 161–181.
- [16] J. Dewey, *Democracy and Education*. New York: The Macmillan Company, 1922.
- [17] S. Büyüktaşkapu *et al.*, “Learning by doing: An example application of “goal-based scenario” approach,” *Education and Science*, vol. 37, no. 165, pp. 293–306, Jan. 2012. [Online]. Available: <https://www.researchgate.net/publication/286171061>.
- [18] J. S. Bruner, *Toward a Theory of Instruction*. Harvard University Press, 1966.
- [19] L. Bot *et al.*, “‘learning by doing’: A teaching method for active learning in scientific graduate education,” *International Journal of Phytoremediation*, vol. 30, no. 1, pp. 105–119, 2005, ISSN: 15497879. DOI: 10.1080/03043790512331313868.
- [20] A. Vescan, “Does learning by doing have a positive impact on teaching model checking?” In *EASEAI 2019 - Proceedings of the 1st ACM SIGSOFT International Workshop on Education through Advanced Software Engineering and Artificial Intelligence, co-located with ESEC/FSE 2019*, Association for Computing Machinery, Inc, Aug. 2019, pp. 27–34, ISBN: 9781450368520. DOI: 10.1145/3340435.3342717.
- [21] L. Stuchlíková *et al.*, “Virtual Reality vs. Reality in Engineering Education,” in *2017 15th International Conference on Emerging eLearning Technologies and Applications (ICETA)*, Sary Smokovec, Oct. 2017. DOI: 10.1109/ICETA.2017.8102533.
- [22] J. Wolfartsberger *et al.*, “Authoring-by-Doing: Animating Work Instructions for Industrial Virtual Reality Learning Environments,” in *Proceedings - 2020 IEEE Conference on Virtual Reality and 3D User Interfaces, VRW 2020*, Institute of Electrical and Electronics Engineers Inc., Mar. 2020, pp. 173–176, ISBN: 9781728165325. DOI: 10.1109/VRW50115.2020.00038.

- [23] FIA Formula E, *Formula E and Virtually Live launch live ghost racing game*, Jun. 2018. [Online]. Available: <https://www.fiaformulae.com/en/news/2018/june/formula-e-and-virtually-live-launch-live-ghost-racing-game/>.
- [24] J. Hamari *et al.*, “Does Gamification Work? – A Literature Review of Empirical Studies on Gamification,” in *2014 47th Hawaii International Conference on System Sciences*, IEEE, Jan. 2014, ISBN: 978-1-4799-2504-9. DOI: 10.1109/HICSS.2014.377.
- [25] A. Domínguez *et al.*, “Gamifying learning experiences: Practical implications and outcomes,” *Computers & Education*, vol. 63, Apr. 2013, ISSN: 03601315. DOI: 10.1016/j.compedu.2012.12.020.
- [26] N. Ashtari *et al.*, “Creating Augmented and Virtual Reality Applications: Current Practices, Challenges, and Opportunities,” in *Conference on Human Factors in Computing Systems - Proceedings*, Association for Computing Machinery, Apr. 2020, ISBN: 9781450367080. DOI: 10.1145/3313831.3376722.
- [27] D. Vergara *et al.*, “On the design of virtual reality learning environments in engineering,” *Multimodal Technologies and Interaction*, vol. 1, no. 2, Jun. 2017, ISSN: 24144088. DOI: 10.3390/mti1020011.
- [28] S. McKenney *et al.*, *Conducting Educational Design Research*. London: Routledge, 2019, ISBN: 9780203818183.
- [29] T. D. Cochrane *et al.*, “A DBR framework for designing mobile virtual reality learning environments,” *Australasian Journal of Educational Technology*, vol. 33, no. 6, Nov. 2017, ISSN: 1449-5554. DOI: 10.14742/ajet.3613.
- [30] A. Mader *et al.*, “A Design Process for Creative Technology,” in *Conference of Engineering and Product Design Education*, University of Twente, Enschede, Sep. 2014.
- [31] M. Karam, “A framework for research and design of gesture-based human-computer interactions,” Ph.D. dissertation, University of Southampton, Southampton, Oct. 2006.
- [32] H. J. Kang *et al.*, “A Comparative Analysis of 3D User Interaction: How to Move Virtual Objects in Mixed Reality,” in *Proceedings - 2020 IEEE Conference on Virtual Reality and 3D User Interfaces, VR 2020*, Institute of Electrical and Electronics Engineers Inc., Mar. 2020, pp. 275–284, ISBN: 9781728156088. DOI: 10.1109/VR46266.2020.00047.
- [33] V. Weser *et al.*, “Making the Visual Tangible: Substituting Lifting Speed Limits for Object Weight in VR,” *Presence*, vol. 27, no. 1, pp. 68–79, 2018. DOI: 10.1162/PRES{_}_a{_}_00319. [Online]. Available: http://direct.mit.edu/pvar/article-pdf/27/1/68/1868890/pres_a_00319.pdf.
- [34] E. Bäckström, “Do you even lift? An exploratory study of heaviness perception in virtual reality,” Stockholm, 2018.

- [35] V. Van Polanen *et al.*, “Visual delay affects force scaling and weight perception during object lifting in virtual reality,” *J Neurophysiol*, vol. 121, pp. 1398–1409, 2019. DOI: 10.1152/jn.00396.2018.-Lifting.
- [36] S. Yoo *et al.*, “VrMove: Design framework for balancing enjoyment, movement and exertion in VR games,” in *CHI PLAY 2018 - Proceedings of the 2018 Annual Symposium on Computer-Human Interaction in Play Companion Extended Abstracts*, Association for Computing Machinery, Inc, Oct. 2018, pp. 295–307, ISBN: 9781450359689. DOI: 10.1145/3270316.3272054.
- [37] M. E. McCauley *et al.*, “Cybersickness: Perception of Self-Motion in Virtual Environments,” *Presence: Teleoperators and Virtual Environments*, vol. 1, no. 3, Jan. 1992, ISSN: 1054-7460. DOI: 10.1162/pres.1992.1.3.311.
- [38] E. Chang *et al.*, “Virtual Reality Sickness: A Review of Causes and Measurements Virtual Reality Sickness: A Review of Causes and Measurements,” *International Journal of Human-Computer Interaction*, vol. 36, no. 17, pp. 1658–1682, 2020. DOI: 10.1080/10447318.2020.1778351. [Online]. Available: <https://www.tandfonline.com/action/journalInformation?journalCode=hihc20>.
- [39] Arvilab, *Combating VR sickness: Debunking myths and learning what really works*, Mar. 2018. [Online]. Available: <https://vr.arvilab.com/blog/combating-vr-sickness-debunking-myths-and-learning-what-really-works>.
- [40] M. Alger, *Visual Design Methods for Virtual Reality*, Sep. 2015.
- [41] S. Fröjdman, “User Experience Guidelines for Design of Virtual Reality Graphical User Interfaces controlled by head orientation input,” Ph.D. dissertation, University of Skövde, Skövde, 2016.
- [42] M. Bernatchez *et al.*, “A study on the impact of spatial frames of reference on human performance in virtual reality user interfaces,” in *2007 IEEE International Conference on Systems, Man and Cybernetics*, IEEE, Oct. 2007, ISBN: 978-1-4244-0990-7. DOI: 10.1109/ICSMC.2007.4414091.
- [43] Y. Weiß *et al.*, “What user interface to use for virtual reality? 2D, 3D or speech-a user study,” in *Proceedings - 2018 International Conference on Cyberworlds, CW 2018*, Institute of Electrical and Electronics Engineers Inc., Dec. 2018, pp. 50–57, ISBN: 9781538673157. DOI: 10.1109/CW.2018.00021.
- [44] H. Kharoub *et al.*, “3D user interface design and usability for immersive VR,” *Applied Sciences (Switzerland)*, vol. 9, no. 22, Nov. 2019, ISSN: 20763417. DOI: 10.3390/app9224861.
- [45] A. G. Sutcliffe *et al.*, “Evaluating the usability of virtual reality user interfaces,” *Behaviour & Information Technology*, vol. 19, no. 6, pp. 415–426, 2010, ISSN: 1362-3001. DOI: 10.1080/014492900750052679. [Online]. Available: <https://doi.org/10.1080/014492900750052679>.

- [46] S. Livatino *et al.*, “Simple Guidelines for Testing VR Applications,” in *Advances in Human Computer Interaction*, InTech, Oct. 2008. DOI: 10.5772/5925.
- [47] J. Brooke, “SUS: A quick and dirty usability scale Display design for fault diagnosis View project Usable systems View project,” Earley, Nov. 1995, [Online]. Available: <https://www.researchgate.net/publication/228593520>.
- [48] Microsoft, *Mixed Reality Toolkit*, May 2019. [Online]. Available: <https://docs.microsoft.com/en-us/windows/mixed-reality/develop/unity/mrtk-getting-started>.
- [49] Unity, *XR Interaction Toolkit*. [Online]. Available: <https://docs.unity3d.com/Packages/com.unity.xr.interaction.toolkit@0.9/manual/index.html>.
- [50] A. Bangor *et al.*, “Determining What Individual SUS Scores Mean: Adding an Adjective Rating Scale,” Tech. Rep., 2009, pp. 114–123.

Chapter 9

Appendices

9.1 Appendix A

9.1.1 Brainstorm

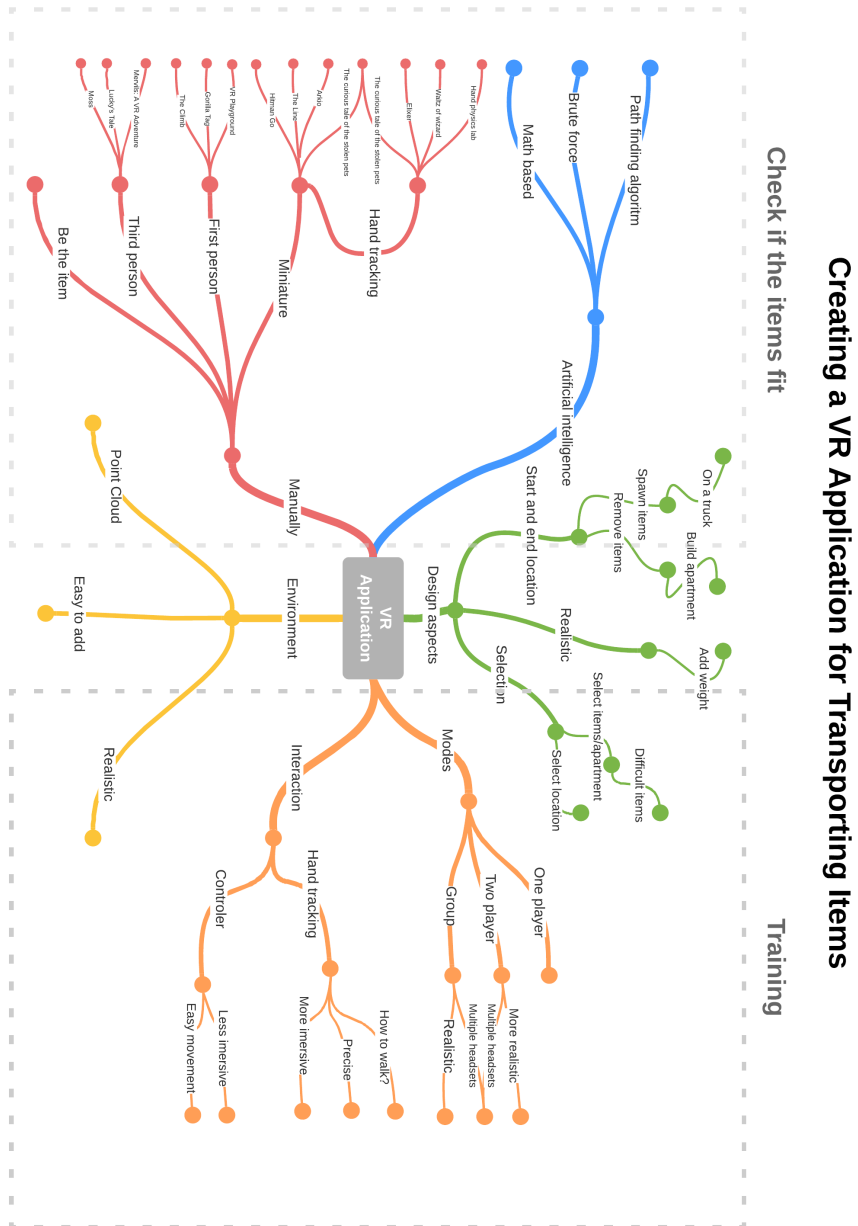


Figure 9.1: Visualisation of combined brainstorm sessions about design aspects.

9.2 Appendix B

9.2.1 User testing

To test the application with user testing a combination of the evaluation methods described by Sutcliffe *et al.* [45] and Livatino *et al.* [46] will be used. The method described by Sutcliffe *et al.* uses a walkthrough evaluation method. This walkthrough is split up into different action stages depending on the kind of walkthrough. This study made use of the 'Goal directed exploration' walkthrough. The evaluation by Livatino *et al.* provides multiple guidelines to take into consideration for setting up an evaluation for VR applications. The guidelines for the evaluation method, setup, participants, forms and test plan have been included in the user testing.

9.2.2 User Testing Process

Each time the application will be user tested, the following process will be performed.

1. Create a safe testing environment for VR.
2. Clean and disinfect the VR headset.
3. Welcome the participant.
4. Explain the research to the participant and tell them what to expect.
5. Let them read the information brochure and consent form (see subsection 9.2.7) and if they agree let them sign the latter.
6. If it is the first time using VR give them some tips and tricks on how to operate in VR.
7. Give the participant instructions on how the user interaction in the application works.
8. If the participant is comfortable proceed to the user testing (see subsection 9.2.3).
9. After the user testing, perform an interview (see subsection 9.2.4) about the application.
10. After the interview ask the participant to fill in the SUS test (see subsection 9.2.5) about the application.
11. Thank the participant for their contribution to the research.

9.2.3 Task list

The following tasks will be asked to be performed by participants. They are asked to think out loud so their thought during the process can be captured on how they think it should be completed.

1. Try to zoom, move and rotate the environment with hand tracking and controllers.
2. Try to pick up items and move them to the endpoint with hand tracking and controllers.
3. Continue this process for all items.
4. Open the 'Place it mode' through the menu.
5. Let an apartment spawn in the environment.
6. Create a small layout of apartments.
7. Open the 'Do it mode'.
8. Try to move around in the 'Do it mode' by teleporting and moving around in real life.
9. Create a collision between an item and the environment.
10. Bring all the items to the endpoint.
11. Close the application using the menu.

9.2.4 Interview

The interview is performed after the user testing of the application. The interview is semi-structured meaning extra questions will be asked when needed.

1. Did you enjoy using the application and why?
2. Do you think the employees of Innovatiehub Salland have an added benefit from this application and why?
3. Were the controls intuitive to use? If not what would be more logical?
4. Did you prefer to use hand tracking or controllers and why?
5. Was this preferred control also the most precise to work with?
6. What would you change/add to the 'Fit it mode'?
7. What would you change/add to the 'Place it mode'?
8. What would you change/add to the 'Do it mode'?

9. Was the interaction with the menu clear?
10. How could the menu be improved?
11. Do you have an understanding of how the items need to be brought inside?
12. Do you have a better understanding of the walking route during a moving day?

9.2.5 SUS test

After user testing the application the participants are asked ten questions. Each question needs to be answered on a Likert scale. Score one corresponds with the answer I strongly disagree and score 5 corresponds with I strongly agree with the statement. The following questions will be asked to the participants.

1. I think that I would like to use this system frequently.
2. I found the system unnecessarily complex.
3. I thought the system was easy to use.
4. I think that I would need the support of a technical person to be able to use this system.
5. I found the various functions in this system were well integrated.
6. I thought there was too much inconsistency in this system.
7. I would imagine that most people would learn to use this system very quickly.
8. I found the system very cumbersome to use.
9. I felt very confident using the system.
10. I needed to learn a lot of things before I could get going with this system.

For each question with an odd number, one point is subtracted from the answer. For each question with an even number, the value of the answer is subtracted from 5. All these new values are added up to a total score. This score is then multiplied by 2,5. This score represents the usability score for that participant. Creating an average of this number for all participants shows the usability score for the application. This score can be compared to industry standards and similar applications. A general score indication can be seen in figure 7.1.

9.2.6 End evaluation

The following questions were performed on the client of this project.

1. To which extend does this application correspond with the initial ideas.
2. What do you believe is going right and why?
3. What do you believe is going wrong and why?
4. Are there certain elements that you would change or remove?
5. Are there certain elements that you would want to add?
6. Do you feel like the municipality, the construction workers and the Innovatiehub can benefit from this?
7. Do you believe the requirements are correctly applied? Which ones do you feel were not correctly applied to the application?
8. Are you overall satisfied with the application?
9. Any additional comments?

9.2.7 Information Brochure and Consent form

Information brochure

Dear reader,

We would like to inform you about the research you have applied to participate in. Innovatiehub Salland wants to build modular apartments in vacant buildings. Before accepting these vacant buildings to be applicable, they want an application in which they can check if all the items can be transported inside this vacant building without changing the existing infrastructure of the building. Next to that, there should be a possibility to train/instruct construction workers in a virtual environment. This research will be on how to develop such an application in virtual reality.

You are asked to participate in the user testing of this application. When you agree to participate you are asked to wear a virtual reality headset (which will be disinfected after each use) and perform tasks in the virtual environment. When performing these tasks you are asked to think out loud, meaning say what you think while performing the tasks. After the using testing, some additional questions about the experience and application will be asked. Using virtual reality can create motion sickness by some individuals. If this occurs, you can remove the headset at any time and quit the user testing. If you have any history of epileptic attacks or are sensitive to motion sickness we advise not to participate in this research. The user testing is estimated around 15 minutes.

Data usage

During the interview, an audio recorder will record the voices of the participant and the researcher, which will afterwards be saved and written down on paper to get a clear overview of the interview. The researcher will also make additional notes during the interview where additional reactions of the interviewee will be noted, such as facial expressions, with the intention to capture unintentional reactions from the interviewee on the information brought by the researcher. Own interpretation of the researcher within the researcher's data collection will be labelled to prevent any miscommunication about what is said during the interview by the participant. All data collected from the participant will be labelled as "participant *number*" to keep the identity of participants anonymously. All data collected during the interview can be used at any moment in this research by the researcher. A possible way of showing data is anonymized quotes of this interview can be used in the thesis.

The data collected in the interview will be analyzed and processed by the researcher within the researcher's Bachelor thesis. The data will only be used within this graduation report on behalf of the University of Twente. The results of the research may be published on the website of the University of Twente and in other research papers from other researchers. The research results may also be used by companies who are interested in developing the device in this research.

The researchers of the project is the only person who can access the collected data. He will store the data in a safe environment to prevent any unwanted spread of information. All data will be kept anonymous to protect the identity of the parties involved. The data will be removed within 2 months after finishing the thesis. When other parties request data of this research, including the participant's involvement, the researcher will only share the data when having permission from all parties involved.

Within 48 hours after the interview, the participant may ask to delete his/her data collected and processed during the research. The data and results will thereby be erased as soon as possible and not be used or mentioned within the research.

Informed consent form

I hereby declare that I have been informed in a manner which is clear to me about the nature and method of the research as described in the aforementioned information brochure. My questions have been answered to my satisfaction. I agree with my own free will to participate in this research. I reserve the right to withdraw this consent without the need to give any reason within 48 hours and I am aware that I may withdraw from the experiment at any time.

I agree to my interview being audio-recorded. Additionally, I agree that anonymized quotes of this interview can be used in the thesis. If my research results are to be used in scientific publications or made public in any other manner, then they will be made completely anonymous. My personal data will not be disclosed to third parties without my express permission. If I request further information about the research, now or in the future, I may contact Tjebbe Treub or his supervisor J. Zwiers.

Contact information

Researcher:

Tjebbe Treub

t.p.treub@student.utwente.nl

Supervisor:

Dr. J. Zwiers

j.zwiers@utwente.nl

If you have any complaints about this research, please direct them to the secretary of the Ethics Committee of the Faculty of Electrical Engineering, Mathematics and Computer Science at the University of Twente, P.O. Box 217, 7500 AE Enschede (NL), email: ethicscommittee-cis@utwente.nl).

Signed in duplicate:

.....

Name participant

.....

Signature

I have provided explanatory notes about the research. I declare myself willing to answer to the best of my ability any questions which may still arise about the research.



Tjebbe Treub

Name researcher

Signature

9.3 Appendix C

The application can be run in different ways. The next few sections explain how to set up the application in multiple ways. To set up the application the following items are required.

- Laptop or PC with graphic card.
- Unity version 2020.3.6.
- Oculus Quest 2 (VR Headset)
- USB-C cable.
- Keyboard and mouse.

9.3.1 Adapting the application

With the right version of Unity installed on a PC, it is possible to open the project folder of the application. No additional software is required to be downloaded when the project opens. In the application, new items and environments can be added. New items that need to be brought inside, need to be placed in the folders under 'Resources'. The folder 'DoItItems' need 3D models that are at a true scale and the folder 'ApartmentItems' need a scale of 0,01. New environments need to be placed inside the Hierarchy itself.

9.3.2 Running the Application on a PC

There are two ways of running the application on a PC to the VR headset. One way is through Oculus Link and the other through Airlink.

Oculus Link

This is the most simple way to connect the VR headset to a PC. Plug the USB-C cable into the PC and the VR headset. In the settings of the VR headset activate Oculus Rift. When the Oculus Rift menu opens, all VR applications can be streamed from the PC to the VR headset.

Airlink

Airlink is used to create a wireless connection between a PC and the Oculus Quest 2. To be able to use Airlink a private WiFi network is required in addition to the previously named items. Connect the PC and Oculus Quest 2 both to the same private WiFi network. Enable Airlink on the Oculus Quest 2 under settings. When enabled, activate Airlink and create the link to the PC. The PC can now wireless stream to the VR headset.

9.3.3 Running the Application on a VR Headset

When the application is fully designed for a task the application can be stored on the Oculus Quest 2. This way after transferring the files from a PC to the headset no other items are required to run the application.

In unity select in the menu under file 'Build Settings'. In the platform settings, switch to Android. Add all the scenes to the window and press 'Build'. Select the folder where you want to store the files. When the building is done transfer this folder onto the VR headset. On the VR Headset go to the folder for unknown sources and select the application. Now the application can be run anytime on the VR headset.

9.4 Appendix D

9.4.1 Code

Item Sequence

```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4 using UnityEngine.UIElements;
5
6 public class Sequence : MonoBehaviour
7 {
8     public GameObject CompletionScreen;
9     public Transform Spawnpoint;
10    public GameObject[] folderObjects;
11    public List<GameObject> objectList;
12    public static bool itemAlive;
13    private int spawnIndex = 0;
14    private bool menuActivator;
15
16
17    void Start()
18    {
19        //Load the 3D models of all items out of the folder
20        apartmentItems into an arraylist
21        folderObjects = Resources.LoadAll<GameObject>("
22        apartmentItems");
23        //Loads the item of the arraylist into a list.
24        foreach (GameObject i in folderObjects)
25        {
26            objectList.Add(i);
27        }
28        itemAlive = false;
29        menuActivator = true;
30    }
31
32    void Update()
33    {
34        //Check if all items are brought to the endpoint.
35        if (objectList.Count == 0 && itemAlive == false &&
36        menuActivator == true)
37        {
38            CompletionScreen.SetActive(true);
39            menuActivator = false;
40        }
41        //Check if there is already an item in the virtual
42        environment.
43        else if (itemAlive == false)
44        {
45            SpawnItem();
46            itemAlive = true;
47        }
48        else
49        {
50            return;
51        }
52    }
53 }
```



```

48     }
49
50     //Spawn a new item on the spawnpoint
51     void SpawnItem()
52     {
53         //Lets the items spawn one by one
54         int i = spawnIndex++ % objectList.Count;
55         GameObject item;
56
57         //Checks if this is the last item in the list
58         if (i == (objectList.Count - 1))
59         {
60             item = Instantiate(objectList[i], Spawnpoint.
61             localPosition, Spawnpoint.localRotation);
62             item.transform.SetParent(transform, false);
63             objectList.Clear();
64         }
65         else
66         {
67             item = Instantiate(objectList[i], Spawnpoint.
68             localPosition, Spawnpoint.localRotation);
69             item.transform.SetParent(transform, false);
70         }
71     }
72 }

```

Do it Mode Sequence

```

1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4 using UnityEngine.UIElements;
5
6 public class doItSequence : MonoBehaviour
7 {
8     public GameObject CompletionScreen;
9     public Transform Spawnpoint;
10    public GameObject[] folderObjects;
11    public List<GameObject> objectList;
12    public static bool itemAlive;
13    private int spawnIndex = 0;
14    private bool menuActivator;
15
16    void Start()
17    {
18        //Load the 3D models of all items out of the folder
19        apartmentItems into an arraylist
20        folderObjects = Resources.LoadAll<GameObject>("doItItems");
21        //Loads the item of the arraylist into a list.
22        foreach (GameObject i in folderObjects)
23        {
24            objectList.Add(i);
25        }
26        itemAlive = false;
27        menuActivator = true;
28    }
29 }

```

```

28
29 void Update()
30 {
31     //Check if all items are brought to the endpoint.
32     if (objectList.Count == 0 && itemAlive == false &&
menuActivator == true)
33     {
34         CompletionScreen.SetActive(true);
35         menuActivator = false;
36     }
37     //Check if there is already an item in the virtual
environment.
38     else if (itemAlive == false)
39     {
40         SpawnItem();
41         itemAlive = true;
42     }
43     else
44     {
45         return;
46     }
47 }
48
49 //Spawn a new item on the spawnpoint
50 void SpawnItem()
51 {
52     //Lets the items spawn one by one
53     int i = spawnIndex++ % objectList.Count;
54     GameObject item;
55
56     //Checks if this is the last item in the list
57     if (i == (objectList.Count - 1))
58     {
59         item = Instantiate(objectList[i], Spawnpoint.
localPosition, Spawnpoint.localRotation);
60         item.transform.SetParent(transform, false);
61         objectList.Clear();
62     }
63     else
64     {
65         item = Instantiate(objectList[i], Spawnpoint.
localPosition, Spawnpoint.localRotation);
66         item.transform.SetParent(transform, false);
67     }
68 }
69 }

```

Exit Application

```

1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class exit : MonoBehaviour
6 {
7     //Closes the application

```

```

8     public void exitApplication()
9     {
10         //Closes the application when runned in Unity
11         UnityEditor.EditorApplication.isPlaying = false;
12         //Closes the application when placed on VR headset directly
13         .
14         Application.Quit();
15     }

```

Menu Activation

```

1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class menuButton : MonoBehaviour
6 {
7     public GameObject menu;
8     private bool toggle;
9
10    //Set menu as inactive in the beginning
11    private void Start()
12    {
13        toggle = false;
14    }
15
16    //Check if the menu needs to be opened.
17    void Update()
18    {
19        ButtonCheck();
20    }
21
22    public void ButtonCheck()
23    {
24        //Check if the menu button is pressed on the controller or
25        //is activated by the handgesture.
26        if (OVRInput.GetDown(OVRInput.Button.Start, OVRInput.
27            Controller.Hands) || OVRInput.GetDown(OVRInput.Button.Start))
28        {
29            //Switch the boolean to the opposite when the button is
30            //pressed.
31            toggle = !toggle;
32        }
33        //Activate the menu with the current boolean for it.
34        menu.SetActive(toggle);
35    }
36 }

```

Object Collision

```

1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;

```

```

4
5 public class ObjectCollision : MonoBehaviour
6 {
7     private Renderer myObject;
8     Color32 objColor;
9
10    void Start()
11    {
12        //When an item is placed inside the environment collect the
13        //color and renderer.
14        objColor = gameObject.GetComponent<MeshRenderer>().material
15        .color;
16        myObject = GetComponent<Renderer>();
17    }
18
19    //When the item collides with something enter this loop.
20    void OnCollisionEnter(Collision collisionInfo)
21    {
22        //If the item hits something with the tag "Disappear" do
23        //the following.
24        if (collisionInfo.collider.tag == "Disappear")
25        {
26            //Delete the item from the environment
27            Destroy(gameObject);
28            //Activate a new item
29            Sequence.itemAlive = false;
30            doItSequence.itemAlive = false;
31        }
32        //If the item hits something with the tag "Collisionable"
33        //make the object red.
34        else if (collisionInfo.collider.tag == "Collisionable")
35        {
36            myObject.material.color = Color.red;
37        }
38    }
39
40    //When the object is no longer colliding, give the item its
41    //original color back.
42    void OnCollisionExit()
43    {
44        myObject.material.color = objColor;
45    }
46 }

```

Load Scene

```

1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4 using UnityEngine.SceneManagement;
5
6 public class loadScene : MonoBehaviour
7 {
8     //Create public code with the scene name as a string input
9     public void LoadScene(string sceneName)
10    {

```

```

11         //open the scene gotten from the input
12         SceneManager.LoadScene(sceneName);
13     }
14 }

```

House Spawner

```

1 using System;
2 using System.Collections;
3 using System.Collections.Generic;
4 using UnityEngine;
5
6 public class HouseSpawner : MonoBehaviour
7 {
8     public Transform truckSpawnpoint;
9     public Rigidbody apartmentPrefab;
10    public BoxCollider TruckBox;
11    private IEnumerator _disableTimer;
12
13    //Public code to start the houseSpawner
14    public void Spawn()
15    {
16        StartCoroutine(DisableTrigger());
17    }
18
19    //The code to let an apartment spawn .
20    void SpawnApartment()
21    {
22        Rigidbody RigidPrefab;
23        //Make sure it spawns at the spawnpoint.
24        RigidPrefab = Instantiate(apartmentPrefab, truckSpawnpoint.
        localPosition, truckSpawnpoint.localRotation) as Rigidbody;
25        //Sets the code as a parent of the spawner.
26        RigidPrefab.transform.SetParent(transform, false);
27    }
28
29    private IEnumerator DisableTrigger()
30    {
31        //Disables the truck so it can not be clicked.
32        TruckBox.enabled = false;
33        //Activate the house spawning code.
34        SpawnApartment();
35        //Wait for 2 seconds.
36        yield return new WaitForSeconds(2);
37        //Activate the truck again, to be clickable.
38        TruckBox.enabled = true;
39    }
40 }

```