Developing a Virtual Reality Tool for the Preparation of Research at the eHealth House at the University of Twente

Giomar Hexspoor
Creative Technology Bachelors Project
September 2019 to January 2020
Supervisor Dr Job Zwiers
Client Dr Miriam Cabrita
Word Count: 21,530
Abstract:

The eHealth House is a simulated living environment at the University of Twente that has recently opened. To help researchers plan their research at this new facility, the goal of this graduation project was “how to create a virtual reality (VR) tool to aid researchers in planning and preparing their research at the eHealth House”.

The created VR tool was made using Unity and features four different furniture layouts to explore. In those layouts, furniture is movable while in VR. The developed application allows the user to observe the virtual environment through the perspective of several CCTV cameras, placed to correspond to their location in the eHealth House. Switching to different scenes can be done using a user interface in the control room. An included tutorial teaches the user all the functions of the program. The researcher can make custom environments with limited functionality using: a program called Sweet Home 3D, the provided eHealth House template, and a provided user instructional guide.

The tool is not without shortcomings. The controls tend to confuse people at first, as they try to figure out what button does what. The physics engine can result in furniture being launched across the room when not placed near perfect after the user picks something up. The way the environment is rendered, despite the furniture and environment being made to a correct to scale, furniture feels smaller, and the room feels larger. Lastly, the initial idea of allowing the user to create their own fully interactive custom environments has not been met. These are the most critical issues to fix.

Despite the shortcomings of the tool, there are upsides. User testing and exhibiting at a demo event show that participants were able to navigate the environment as well as complete multiple tasks. Users found themselves immersed and engaged in the VR application. Even with its flaws, researchers and eHealth House staff alike seem to be positive about the future possibilities of such a tool. Future research and development can lead to an application that makes all custom designs interactive, make the tool more useful for showcasing the eHealth House’s capabilities, and potentially testing patients inside VR.
Acknowledgement:

It should be noted that the idea for this project was proposed by the client, not the author of this paper. Likewise, any ideas taken from other sources, including figures, quotations, and the like have their work attributed to their rightful owners.

I would like to thank my supervisor Job Zwiers and client Miriam Cabrita. I know I am not the easiest student to work with, yet you showed patience, guided me through the process, and clarified the same thing time and time again. Thank you very much.

Thank you to Alfred de Vries for allowing me to use university equipment extensively. I would also like to thank all people who took time out of their day to help me with interviews, user testing, and solving problems.

Lastly, I would like to thank all the people (and you know who you are) who had to put up with me during the graduation semester. Be it with an onslaught of terribly timed questions, being the victim of my hours of complaining, or dealing with my mood swings. Thank you. Without you, I would not be where I am today.
Table of Contents:

ABSTRACT: ............................................................................................................................. II

ACKNOWLEDGEMENT: ........................................................................................................ III

TABLE OF CONTENTS: ......................................................................................................... IV

TABLE OF FIGURES: ........................................................................................................... VI

1  INTRODUCTION: .................................................................................................................. 1

2  STATE OF THE ART: ......................................................................................................... 3

2.1  STUDIES IN THE REAL ESTATE MARKET: ................................................................. 3
2.2  TOOLS USED TO MAKE VR ENVIRONMENTS: ......................................................... 5
2.3  CYBERSICKNESS: .......................................................................................................... 5
2.4  USER INTERFACE: .......................................................................................................... 8

3  IDEATION: ........................................................................................................................... 10

3.1  IDEA 1: ........................................................................................................................... 11
3.2  IDEA 1.5: .......................................................................................................................... 11
3.3  IDEA 2: ........................................................................................................................... 12
3.4  IDEA 2.5: ........................................................................................................................ 13
3.5  CHOSEN IDEA: .............................................................................................................. 14
3.6  TOOLS TO BE USED: .................................................................................................... 14
3.7  IMAGINED WORK PROCESS: ...................................................................................... 15

4  SPECIFICATIONS: ............................................................................................................ 17

4.1  INTERVIEW: ................................................................................................................... 17
4.2  REQUIREMENTS: .......................................................................................................... 18
4.3  SPECIFICATIONS SHEET: ............................................................................................ 22

5  INITIAL APPROACH: ....................................................................................................... 25

5.1  PLANNED WORK PROCESS: ....................................................................................... 26
5.2  THE PROBLEM: ............................................................................................................. 28

6  ACTUAL PRODUCT: .......................................................................................................... 32

7  REALIZATION: .................................................................................................................. 42

7.1  PRODUCTION: ............................................................................................................... 42

8  EVALUATION: ................................................................................................................... 55

8.1  COMPARING THE TOOL TO THE REQUIREMENTS: .................................................. 55
8.2  USER TESTING: ............................................................................................................. 57
8.3  PRESENTATION TO EHEALTH HOUSE RESEARCHERS: ........................................... 66
8.4  EVALUATION SYNOPSIS: ............................................................................................ 67

9  IMPROVEMENTS: ............................................................................................................ 69

9.1  DEFINITE IMPROVEMENTS: .......................................................................................... 69
9.2  IMPROVEMENTS TO SWEET HOME 3D INTEGRATION: .......................................... 70
9.3  IMPROVEMENTS TO SINGULAR APP: ........................................................................ 70
Table of Figures:

Figure 1: Traditional forms of showcasing environments (Juan et al., 2018, p. 5) ........................................4
Figure 2: “Arm–Cycling: When both triggers are engaged, users are translated forward in the direction they are facing according to the absolute value of the displacement of the controllers.” (Coomer et al., 2018, p. 2) ................................................................................................................................. 7
Figure 3: "ArmSwinger Mode" (Loup & Loup-Escande, 2019, p. 1272) ............................................................ 8
Figure 4: Flowchart indicating a possible process of using the VR tool .......................................................... 16
Figure 5: A table with mesh collider (green lines) __________________________________________________________ 29
Figure 6: Interaction between a pillow and an armchair both of which have a mesh collider ............................ 30
Figure 7: User Controls – Original picture by Oculus (2020) ........................................................................... 34
Figure 8: A screenshot of one of the camera perspectives, allowing the user to see the eHealth House as the cameras do .......................................................................................................................................... 35
Figure 9: A top-down view of the eHealth House tutorial, teaching users how to use the controls – Scene 0 ................................................................................................................................................. 36
Figure 10: A top-down view of the Empty eHealth House scene created. Can also be used to import custom settings – Scene 1 ...................................................................................................................................... 37
Figure 11: A top-down view of the eHealth House with a moderate amount of furniture. Also, an example of what a researcher might plan their setting like – Scene 2 ........................................................................... 38
Figure 12: A top-down view of the eHealth House containing more furniture. Also an example of what the eHealth House might look like inhabited – Scene 3 .................................................................................... 39
Figure 13: A top-down view of the eHealth House containing a grand piano. Also, functions as a reminder of the usability and advantages of the tool – Scene 4 .................................................................................... 40
Figure 14: A top-down view of both the empty scene and the outside world that can be seen from the windows ......................................................................................................................................................... 41
Figure 15: A armchair and pillow with custom hitbox colliders (green lines) .................................................. 44
Figure 16: Chairs and table with custom made hitbox colliders (green lines) .................................................. 45
Figure 17: Teleportation Settings ....................................................................................................................... 48
Figure 18: OVR Player Controller and its children ............................................................................................ 49
Figure 19: The puzzle in the tutorial ................................................................................................................ 51
Figure 20: The UI as displayed on the Control Room Wall .................................................................................. 52
Figure 21: A view of the garden from the eHealth House ................................................................................... 53
Figure 22: Sweet Home 3D working space ......................................................................................................... 80
Figure 23: Category Folder way of displaying furniture .................................................................................... 81
Figure 24: Searchable List way of displaying furniture ...................................................................................... 81
Figure 25: File -> Preferences .......................................................................................................................... 82
Figure 26: Preferences List .................................................................................................................................. 82
Figure 27: The modify furniture window ........................................................................................................ 83
Figure 28: Different possible poses using the modify posture function .............................................................. 85
Figure 29: Furniture Materials window ........................................................................................................... 86
Figure 30: Example of selected items ............................................................................................................... 87
Figure 31: Unity Workspace .............................................................................................................................. 88
Figure 32: Example of what the inside of the imported folder can look like ..................................................... 88
Figure 33: Inspector settings ............................................................................................................................ 89
1 Introduction:

As technology develops, the way researchers perform, and conduct research develops as well. An example of this can be seen at the University of Twente, where recently, the Tech-Med Center was renovated. With it comes a high-tech simulated living environment (SLE) known as the eHealth House (eHH). An SLE is a facility that can be used by researchers to test and observe patients in a realistic environment. Here, the patients face situations that realistically reflect what they would experience and face while in the environment that is being simulated. Making it beneficial in learning and studying how people interact in the desired environment (Brewin et al., 2015; Hecox et al., 1994; Mole, Scarlett, Campbell, & Themessl-Huber, 2006; Richardson, Law, Wishart, & Guyatt, 2000). In the case of the eHH, this environment consists of two parts. The first part is an apartment consisting of a living room, kitchen, bedroom, and bathroom; the second part is a meeting room. Both parts can have their interior redecorated to simulate different real-life settings. Additionally, the environment is constructed around a raised, central control room in which the researchers can observe and collect data from the environment using a wide variety of sensors, tools, as well as two-way mirrors.

When researchers are planning the setup of their experiment at the eHH, they need to create an interior of the eHH to suit the needs of their patients and themselves. Their final setup should be functional so that the patient can still properly move around and interact with the environment. It should also be made in such a way that the researchers can properly observe everything from the control rooms and CCTV cameras; all the while being reflective of researchers’ desired setting. However, without the proper tools in place to let researchers visualize their setup, the created designs can contain errors. Mistakes can lead to a loss of productivity, money, as well as invalidate data. Further, such errors are more prone to happen to individuals who do not have easy access to the eHH; and thus, cannot visit it often to reference in their design process.

From the above, it becomes possible to derive the research question and goal of the graduation project: “how to create a virtual reality (VR) tool to aid researchers in planning and preparing their research at the eHealth House”. This paper covers the creation of a VR tool which allows users to navigate and interact with the eHealth House in such a way that it can help their planning and preparing phase. The environment can be explored with the use of a head-mounted display (HMD), specifically the Oculus Rift S.

The created VR environment features four different furniture layouts that can be explored. Any furniture that should be movable, not fastened to the walls, floor, or ceiling, can be moved in VR by the
user. Further, the developed application allows the user to observe the virtual world through the perspective of several CCTV cameras, placed to correspond their actual location in the eHH. Overall, an immersive experience and environment are created, allowing the user to interact and explore an accurate replica of the eHH. It also allows for the importation of custom environments made by the end-user, permitting one to explore an environment made to specific requirements. Custom environments do lack functionality as objects cannot be picked up nor moved.

The tool can thus be used by researchers to help visualize different setups, allowing the researcher to explore them in VR. The immersion and engagement provided by VR give users an insight into the environment that is not available using traditional methods (Juan, Chen, & Chi, 2018). This way, it becomes possible to create and test different environments cheaply and relatively quickly. The created environment can be seen from the perspective of the CCTV cameras, control room, and the researcher’s perspective anywhere in the environment. This tool can also be used anywhere, meaning that researchers no longer need to go to the eHealth House to walk through its rooms. Such a tool can have the ability to make the development and preparatory stages of research cheaper, less laborious, and less prone to errors.

Commissioned by the Personalized eHealth Technology Research Program, the tool enhances the toolkit that the University can provide to researchers who are using the SLE. Not only is the created application a development build of an asset to help researchers plan and prepare their research at the eHH, but it also provided valuable insight into important questions and factors that need to be considered for further development of this or likewise tools.

This paper will provide a holistic overview of the design process underwent and explore the steps taken to develop the created tool. Followed by an analysis of the conducted user testing and recommendations for improvements, the paper will be finalized by looking at the possibilities of future work.
2 State of the Art:

This section will look at the state of the art. Due to a lack of literature on VR tools on designing SLEs, conducted research looked at VR tools in the real estate market, as that overlaps in many ways with the ideas at hand. The state of the art will also explore several tools used in the field for designing VR homes. Followed up with research into cybersickness and finalized with information regarding user interfaces.

2.1 Studies in the Real Estate Market:

The real estate market is one in which virtual reality has started to become more used in order to provide tours of residences to clients. These tours highlight details and information that would otherwise not be possible. It is thus a great starting point to discover the usefulness of virtual reality in visualizing a house and furniture compared to existing methods. In order to pique interest, agents would often present clients with images, blueprints, and 3D models of a house to get them more excited and narrow down options; especially when visiting all the houses is not a possibility. Agents can now visit homes with the use of VR.

One study looked at using VR to allow the user to create the interior design of a venue for hosting special events. The user can make this in a digital VR environment as they might not have access to the site for different reasons such as construction or distance. They can manipulate the environment by adding, removing, moving, and cloning furniture in the venue. They are then able to view this in VR by using a head-mounted display (HMD), namely, the Oculus Rift DK2, as well as Razer Hydra controllers. The program was made using Unity (Chenechal, Lacoche, Martin, & Royan, 2015).

Another study looked at pre-housing sale in Taiwan. Pre-housing sales are when companies sell real estate, before the construction of the sold item has begun or finished. Virtual reality can be used in order to let people experience the building and its rooms without construction being finalized. In this study, a VR application was made to be used with a Samsung Gear VR and the Samsung Galaxy S7- which when combined form an HMD. The model of the building was made using Sketchup, and the virtual reality environment was rendered using Lumino6 and Enscape. The virtual environment can be explored using the web-app 720YUN (Juan et al., 2018).

The same study then evaluates their VR tool by comparing it to more traditional forms of presenting the user with information of the house. Blueprints, images of renders, and a physical scaled model of the
building as can be seen in Figure 1: Traditional forms of . Using Technology Acceptance Model questionnaires, the results found that while there is no significant difference in the ‘ease of use’, there are significant differences in “perceived usefulness”, “users’ attitude”, and “users’ intention”. Meaning that the users managed to get more information that they found useful, their attitude towards the house bettered and that there is a higher intention to use this technology, with the use of the created tool. Thus, showing that VR is beneficial to use for these purposes (Juan et al., 2018).

Figure 1: Traditional forms of showcasing environments (Juan et al., 2018, p. 5)

The study by Ozacar, Ortakci, Kahraman, Durgut, and Karas (2017) covered the process of creating a tool for similar purposes as the previous papers. Ozacar et al. (2017) produced an interactive VR environment; not only allowing for the user to walk around but also alter elements such as the material and texture of the floor and walls. This application was created by combining multiple programs. First, Sweet Home 3D (SH3D) was used to make the model of the house. This model was then imported to Unity, in which the VR environment was created. An HTC Vive (also an HMD) was used in combination with a controller to manipulate and interact with the environment (Ozacar et al., 2017).

Preliminary tests were run in order to test the first impressions of the VR tool. A vast majority (15 out of 18) of the participants had no prior experience with VR. No reference was made to users experiencing difficulty. However, it was mentioned that 2 of the 18 participants would have preferred using arrow keys to walk around instead of the provided teleporting function. Participants were asked to provide their opinions on several questions on a scale from 1 (lowest) to 5 (highest). The questions: “Did you feel like you were walking in a real house?” scored a 4.56; “Did you like user interaction design?” scored 4.39; “How good was the graphics quality?” scored a 4.00; lastly, “would you buy a home after taking such a VR tour?”
scored a 3.94 (Ozacar et al., 2017). The conclusion of the paper states that the use of a VR tool provides the user with a beneficial form of interaction with the living environment, and provides them with a good insight into what the environment could look like in real life (Ozacar et al., 2017).

2.2 Tools Used to Make VR Environments:

Analyzing how VR environments are made for the real estate market, two main methods seem to be predominant. The first is based in VRML (Virtual Reality Modeling Language) working in unison with Java. The second method uses CAD software and imports the result to Unity 3D.

The first of the methods mentioned was VRML in combination with Java. This is seen in several VR applications (Qi, 2012; Vosinakis, Azariadis, Sapidis, & Kyratzi, 2007). Further, Kun and Zong (2009) state that VRML is the way VR applications are usually created in the real estate industry. The accuracy of this statement is juxtaposed to the use of CAD software and then importing the result to Unity, as shown by other studies. (Chenechal et al., 2015; B. Deaky & Parv, 2017; Bogdan Deaky & Parv, 2018; Ozacar et al., 2017; Pejić, Krasic, Veljković, Sakan, & Rizov, 2016; Yu, 2011) The use of Unity seems to be more prevalent than the use of VRML.

Further, one does not necessarily need to use CAD software to integrate Unity. Ozacar et al. (2017) does not use CAD software and instead uses a program called “Sweet Home 3D”, an application that is free and allows the modelling of houses in a matter of minutes- even for those with little to no experience with the program. CAD software is specialized and often takes longer to learn (Vosinakis et al., 2007). In SH3D, furniture can be added to the house with a drag and drop system. The created model can then be imported to Unity, simplifying a large part of the process.

2.3 Cybersickness:

Since the user will find themselves in a virtual environment, there needs to be a way they can navigate the environment. The downside of VR is that sometimes VR causes cybersickness; resulting in people to experience symptoms similar to motion sickness. This is thought to be because the disassociation the user experiences while in VR; it seems like they are moving while their body remains in place. Resulting in symptoms such as, but not limited to, headaches, nausea, and vertigo. Furthermore, age, gender, illness, and other factors can cause this or work as a catalyst for this effect. (Davis, Nesbitt, & Nalivaiko, 2014; Joseph J. LaViola, 2000). Experiencing cybersickness while using a VR tool discourages further use and should thus be minimized.
With the nature of HMDs, the user will be able to walk around for some limited navigation as one's movement is translated. Moving one meter in real life moves the player one meter in VR. Movement translation is not enough to explore the entirety of the virtual eHealth House. A secondary form of movement is thus needed; known as locomotion. Different forms of locomotion can cause different amounts of cybersickness (Christou & Aristidou, 2017; Clifton & Palmisano, 2019; Coomer, Bullard, Clinton, & Williams-Sanders, 2018; Loup & Loup-Escande, 2019). Hence in order to lower the chances of a user getting motion sickness a way of traversing the environment, it is essential for the right type of locomotion to be chosen. Hence, it is crucial to investigate the different forms of locomotion that are available.

Literature does not seem to be unanimous when considering the best type of locomotion to use. Coomer et al. (2018) propose a movement technique of their creation called “arm cycling”. In this, the user makes movements somewhat like the breaststroke to move forwards, as illustrated in Figure 2. Coomer et al. (2018) note that there is no statistical difference between this method and teleportation (instant relocation to the desired location, which can be selected with the controller) regarding causing cybersickness. For other factors, namely, the amount of disorientation caused by the use of this form of locomotion, arm cycling placed higher than teleportation. Another study showed that teleportation still had the possibility of causing cybersickness, with results varying greatly per participant; however, this was also true for other forms of locomotion (Clifton & Palmisano, 2019).
While some studies indicate that there does not seem to be a difference in individual form of locomotion, others juxtapose this view. Christou and Aristidou (2017) state that while comparing teleportation to two steering techniques (in this case, pointing and gaze focus-looking in a specific direction for a while, results in movement is said direction), teleportation seems to cause the least cybersickness. Likewise, Loup and Loup-Escande (2019) report that teleportation results in less cybersickness then arm swinging (locomotion in which moving one's arms as one would do when running results in movement, as shown in Figure 3). Teleportation is also backed up by Buttussi and Chittaro (2019) as the form of locomotion that causes the least cybersickness.
2.4 User Interface:

Some literature was referenced in order to help with the decision process of helping with UI choices. Kim and Kim (2017) suggest that that information will be easier to process in VR if the UI that displays it is attached to either the controller that they are holding or integrated into the player’s body. Elements such as waypoints or explanations of items would need to be centre frame, with an indicator line drawn to the item it is referring. For menus, using a radial format based around the controller seems to produce the best results. If it is a very sophisticated menu, the use of laser pointers works better than having to touch the menu item with the controller; while more efficient, it is at the expense of immersion into the environment (Kim & Kim, 2017).

Further research had to be derived from theses. Sanders (2018) stipulates that an improper user interface can prevent a user from understanding the primary purpose of the program. Further, at times it might be most beneficial to combine different types of interfaces in order to get the results in regards to user experience and results. Lastly, Fröjdman (2016) provides guidelines for the creation of user interfaces when using one’s head as the form of selecting items; by focusing on a particular point for a certain length of time, it gets selected. When using this form of interaction, it is best to place the UI between 75 centimeters and 3 meters. It is also important to have the interface within the bounds of 60 degrees horizontally, equidistant from the heads forward facing position, hence no more than 30 degrees in either direction. With the vertical bounds being at 32 degrees, divided with 20 degrees upwards, and a maximum
of 12 degrees downwards. The thesis provides many other aspects that are relevant and important to consider if a gaze-controlled UI is used.
3 Ideation:

Upon the first meeting with the client, the initial concept was to create an application that would allow researchers to see the eHealth House in virtual reality. Researchers could use such a tool if they wanted to know what the eHH looks like. Such a tool would be beneficial for both people who have never seen the eHH before, allowing them to see it. For people who have been there more often; they can use the environment to explore it more in-depth or recall specific details about it. A comparison was made to a common strategy used by people designing a physical environment, to draw it on a piece of paper, and then use smaller, cut out pieces of paper as furniture. Moving the smaller pieces of paper around would allow for different setups to be created inside this environment.

A VR tool could integrate features that enable the user to create different settings for the eHH, allowing the researcher to create custom environments and explore them in VR. This could be beneficial to test different furniture setups quickly. The VR tool would allow the user to get a good understanding of how the environment will look with their created setup. With the ability to walk through the virtual creation, the user would be able to see the environment from different perspectives. Being able to see through the lenses of the cameras installed in various places around the eHH would be beneficial to this goal as well.

Scenario:

As a scenario, take a hypothetical researcher called Dr Stone, who is considering doing his research in the eHH. Dr Stone works at a university on the other side of the country, and he has never seen the eHH but wishes to learn more about it. After talking to the University of Twente, he gets sent the files to this VR application which he can run using the facilities at his university.

He uses the VR tool to explore the eHH, and it piques his interest. He decides it is worth visiting the eHH in person to see more of what this environment has to offer.

After his visit, he is determined to use the eHH. A bit further into the research process, he and his colleagues are wondering about several potential setups for the experiment they want to conduct. They have designed several layouts but would like to see what they would look like in person. However, it is a lengthy process to travel cross country, and an even more laborious, lengthy, and costly process to change the entire layout of the eHH multiple times to see which one they like best. That is when Dr Stone remembers the VR tool he used, also recalling it was possible to create custom environments using it.
Dr Stone opens the application, and in less time then it would have taken him to travel to the University of Twente, has managed to create three different environments to explore for him and his colleagues. As they explore this environment in VR, they see several flaws in some of their choices. Certain angles from the CCTV cameras have poor vision over the room due to selected furniture, and in other places, there is not enough space for the patient to properly move around.

They soon realize that scene two that they created is the most feasible, but they also note that it is still somewhat lacking. In VR, they pick up furniture and move it around to make it better until they are finally set on a design to use in their experiment.

Satisfied with their designed environment, they take note of the setup and continue planning their research around their created design.

3.1 Idea 1:
It would be possible to create such a tool using Unity. Allow the user to select furniture from a menu and place it inside the eHH. Basically, taking the pen and paper idea and making a digital version of it. The cut put pieces of paper representing furniture gets replaced with furniture items on the menu.

The user would be able to select from a wide range of furniture and place it how they liked. A snapping feature can be used in order to ensure that the items can be placed directly to the walls. In many ways, the envisioned functionality of creating the environment was very reminiscent of The Sims. A significant difference is that in The Sims, everything is done in a 3D perspective. This tool would utilize a 2D top-down approach. When done designing, the researchers would be able to run their created environment inside VR.

Difficulties that would be faced here is with the implementation of certain features. Such a program would require a way to rotate, adjust the height, and scale furniture, which could be challenging to implement. A considerable struggle would be dedicated to creating a proper and fully functional UI.

3.2 Idea 1.5:
In much a similar vein as idea 1, idea 1.5 would utilize a program discovered during the state-of-the-art research process. Sweet Home 3D, as used in the article by Ozacar et al. (2017), is an application that lets the user create their home, and add furniture to it, at their discretion. It comes with features allowing the user to move, rotate, scale, elevate, and even change the material of objects. All of this can
be exported to OBJ format and then imported to Unity. This application solves many of the issues with idea 1; however, it does come with new challenges.

The downside with this is that Sweet Home 3D does have a rather peculiar user interface. The 3D rendering of the image is always centred around a single point that cannot be changed; making it challenging to get the perspective from certain angles of the created environment. The way of altering the scale, rotation, or elevation of furniture can be quite tricky. It is only possible to do this from the 2D scene but to see how it looks; one needs to reference the 3D scene; likely from a terrible angle. To get things placed just right becomes somewhat tricky. It is also more accessible to change the scale, rotation and elevation via manual typed input because using the 2D interface for these functions can be challenging to use.

Another downside of this is that the user will need to learn to use two programs instead of one: Unity and SH3D. They will also need to learn how to export and import items between the two. Resulting in a steeper learning curve for the user, which can result in people being less likely to use it.

Additionally, Unity needs to differentiate between the imported items, which can be a challenging process. A script would need to be written to differentiate the objects based on what piece of furniture they are, as well as add the required components to these pieces of furniture. This will be a complicated and laborious process to do.

3.3 Idea 2:

Another proposed project had to do with making a business card that made use of AR (augmented reality). Leading to another idea; creating a product that takes a physical representation of the eHealth House and transforms it into a digital environment. Using the idea mentioned earlier of cut out pieces of paper representing furniture upon a drawing; create a tool to take that information and turn that into a virtual environment. Taking a big sheet of paper with the floorplan of the eHH on it, except in the corners of the page are QR codes, would allow an application that runs an AR program to superimpose an AR version of the eHealth House on it. If the user wants to place down a couch, it would be possible to take a token that is made to scale with a QR code on top as well. The user can place this token inside the floorplan, and the AR app will superimpose a virtual couch on that. Using a multitude of different tokens, custom environments can be made. The user can decorate the environment to scale, not having to worry about if objects can fit in the environment if they fit on the map.
This AR app could work by pointing a phone screen at the floorplan. The user will be able to walk around it and see it from different angles.

Several problems form with the introduction of such a program. The first is that the entire floorplan must be shown in order for the AR environment to be superimposed, as all the QR codes need to be read by the app. This also means that physically zooming in by moving the phone closer will be relatively limited. Zooming will need to be done digitally, such as by pinching one’s fingers on the screen.

Stacking furniture becomes a problem as well. The camera needs to see the top of the token to read the QR code. So, adding a TV on top of a closet, as well as multiple books, lamps, and plants, will become difficult.

Small items will require tokens that cannot be made to scale. If the environment is displayed on an A3 paper, to make a token of a fork or glass to scale would require their tokens to be no more than 3 millimetres on all sides.

While the rotation and position of a token can be easily changed, there is no inherent way of changing their height, unless something is placed underneath it. This means that there need to be pieces made to scale to elevate items to a certain height — a lengthy process for a relatively simple feature. Scaling furniture would have to be done on the mobile as well.

Interaction with the furniture would either be limited to real-life or would require very complicated scripts to be able to interact with the furniture in the AR environment.

Lastly, adding a way that will allow users to create custom furniture could be difficult. The user would not only need to create the model and import it to the program but also create a new QR code to go with their custom piece of furniture. They will likely need to manually update the phone app and reinstall or update it on their phone.

3.4 Idea 2.5:

This is nearly identical to idea two, except that the data is used to render a 3D environment that can be explored using VR instead of AR. This would also mean that the computer would require a webcam aimed at the floorplan. It would allow the user to design the room in real life, in much the same way they would by using the standard old method, but then be able to experience it in VR.
The problems this idea creates would be identical to the problems from idea 2. An additional problem would be that the program would not have to superimpose an image on a QR code, but rather create an environment based on the locations of the QR codes. It would need to take the received image, and from that create the environment to explore in VR, a more complicated process.

3.5 Chosen Idea:

From the different ideas, idea 1.5 seems to be the most viable. This conclusion was reached by process of elimination. Idea 2 and 2.5 come with too many complexities to get to properly function, making it an unrealistic undertaking in the timeframe of this project. Ranging from fixing issues related to the tokens’ scale and creating the program itself.

Idea 1 comes with many complications, requiring many systems to be created that come naturally to Unity. This is, of course, assuming that a final application gets produced as a stand-alone program, and not that the Unity editor will be used. Moreover, while it is possible to ultimately develop this tool from scratch using Unity, resulting in a single program, that might not be the most efficient way to achieve the end goal, considering that doing so would be an extensive project. As stated by Humm and Ossanloo (2016, p. 1) “software development today means, to a large extent, integrating existing software components.”

Many issues that idea 1 has gets solved by 1.5. Having an existing application that allows the user to model the environment to their liking and export it to Unity in one go can be very beneficial. One larger script would have to be written to add the required components to furniture, but other than that it should not pose as many problems as idea 1 would have.

3.6 Tools to be used:

The virtual reality environment will be made using Unity. Not only is Unity a program which is useful for making Virtual Reality Environments, as can be seen by multiple studies referenced in section 2.2, it is also a tool that is taught in Creative Technology. There is thus familiarity with the program and how it works. Further, Unity is an industry-standard game engine (Ramey & Panter, 2015; Yang & Jie, 2014).

Unity can be used to make a 3D environment, but also to take that environment and integrate the required components for it to become explorable with the use of VR technology. Creation of a 3D environment is thus required. As mentioned, Unity can be used to create environments; it does come with a plethora of limitations. Maya allows for complete environments to be modelled to the users' wishes, although doing so is a lengthy process. Comparatively, Sweet Home 3D allows for house designs, including
furniture, to be quickly created and exported to OBJ format, meaning that Unity process it. Making the environment with this tool is beneficial.

There are two main reasons Sweet Home 3D was chosen. First, alternativeto.net, a website that can be used to discover software similar to the provided application (in this case Sweet Home 3D) (Humm & Ossanloo, 2016; Johansson & Olausson, 2019), was used to discover different programs. A plethora of likewise programs was recommended. One can click on any of them, to provide a similar list for the newly selected program. Appendix 1 – List of available software: contains a list of several of these programs. Since the market seems to be oversaturated with such programs, making use of existing software is a smart decision. Alternativeto.net also mentions that of the listed programs, “Sweet Home 3D” is the only open-source program from that list. Open-source software is released in such a way that it allows for the source code to be viewed and edited by anyone (Crowston, Wei, Howison, & Wiggins, 2008), meaning that need by, SH3D can be changed to the project’s needs.

Secondly, there exists literature using this program. Ozacar et al. (2017) also used Sweet Home 3D to create the interior of their environment. Seemingly, from the available software, SH3D is the only open source one, and has citations in existing literature, making it the right candidate for being the selected tool of use to create the virtual eHH.

3.7 Imagined Work Process:

First using Sweet Home 3D, the user will be able to plan out their design of the eHH by dragging furniture onto a two-dimensional plane. Rearranging furniture to their liking. Placed furniture can be edited according to the researchers wants. This is achieved by double-clicking the item. Size, material, alleviation, rotation, and mirroring are some elements that can be changed. The researcher will be able to receive immediate feedback on their design, as Sweet Home 3D will also display their creation in three-dimensions. If the environment is as desired, it can be exported into OBJ format, allowing the importing into Unity 3D. Once imported, the environment should be able to be experienced in virtual reality with minimal input from the researcher. The researcher can then experience the environment in VR. If the environment is not as desired, they can decide to change the setup in Sweet Home 3D until they find it satisfactory. This imagined work process is laid out in Figure 4.
Figure 4: Flowchart indicating a possible process of using the VR tool
4 Specifications:

Before the construction of the tool can be started, there must be a clear set of specifications to function as the requirements of what the tool needs to be able to do. The requirements and necessities for this tool are based on three aspects. An interview with Hermans (2019), the content coordinator of the eHH; the ideas from section 3; and the needs and desires set forth by the client.

This section of the paper is also divided into three subsections. The first subsection provides more information on the interview. The second lists the produced requirements and is thus the cultivation of the three aspects listed above. It will provide further detail and explanation for the most critical requirements. The last subsection is dedicated to a specification sheet, that describes what the product is and how it is supposed to function.

4.1 Interview:

To ensure that the desires of the users are appropriately considered in the formulation of the requirements, an interview was conducted with Mathilde Hermans; the content coordinator of the eHealth House (Hermans, 2019). She will be able to provide more information on not only what features would be desired, but likewise on their importance in the product. She will be able to provide a fresh perspective and bring new ideas to the table.

The interview allowed new perspectives to be explored. Prior the focus had solely been on the development of a tool dedicated to helping researchers plan and prepare their research. The interview highlighted that the desire for a VR tool is not limited to preparation only. Instead, there is also a desire for a VR showcase tool to be taken to conferences. This can then be used to introduce people to the eHH, let them explore the environment, and see the capabilities the eHH has the offer. A showcase tool would focus more on developing one or two highly interactive environments. The interaction could include the ability to interact with smart home features in the virtual environment by the use of a virtual iPad. A fridge which can be opened and contains food and drink with which the user can interact. A fully functional intercom system, and perhaps even phones that ring requiring the user to answer. Many of these highly interactable features are not necessary for a preparatory tool, whose focus is on allowing researchers to plan their research.

The research tool will have less interaction, likely limited to moving objects, opening and closing doors, and potential interaction with lights. While this tool would offer less intractability, it would be
applied to more scenes. The showcase model would have one or two highly interactive scenes; while the research tool will allow for minimal interaction but have that on scenes that the user can create themselves — allowing users to shape the environment to their will and desires. The user gets to choose what goes where, and what the environment looks like, at the cost of not having advanced features such as a showcase tool would have. Other features could be beneficial for a research tool, though, such as adding the ability to measure distances between two points. Further, there would have to be a more significant focus on streamlining the integration of SH3D to one that enables easy integration with Unity.

The client had specified that the main objective of the tool to be a research tool, with possible secondary objectives being showcasing, and hence that premise shall be retained. The interview having consisted of the deliberation of two types of tools and their differences, allows for the distinction between showcase and research to be more clearly established. This should allow for a tool more focused on the researchers to be produced. With careful integration of some showcase features discussed during the interview will allow the tool to become more comprehensive overall. An example is a more comprehensive and inclusive control room for the virtual environment. Initially, the most critical aspect was to create a control room in such a way that enabled the researchers’ in-person view of someone in the eHealth House, as well as perspectives of different CCTV cameras to be seen. However, the interview provided valuable input that making this more extensive can be beneficial for researchers and showcase alike.

4.2 Requirements:

The requirements are what will be expected of the product. Based on information gathered from the state-of-the-art research, the initial idea, consultation with the client, and interview. From this, a product can be made for testing. The requirements will be organized in a table containing two columns, the first being the feature and possibly a little explanation, and the second is the importance it has in the product. The priority of implementation has been based on the MoSCoW method. The table functions as a reference to what must be included in the product, both during and post-development. The requirements and their priority for implementation are displayed in Table 1.

<table>
<thead>
<tr>
<th>Feature:</th>
<th>Priority:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Smooth Integration with SH3D</td>
<td>Must have</td>
</tr>
<tr>
<td>Ability to move around</td>
<td>Must have</td>
</tr>
<tr>
<td>Feature</td>
<td>Requirement</td>
</tr>
<tr>
<td>------------------------------------------------------------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>A secondary form of movement</td>
<td>Must have</td>
</tr>
<tr>
<td>A full model of the eHH</td>
<td>Must have</td>
</tr>
<tr>
<td>Ability to pick up objects (that can be picked up in real life)</td>
<td>Must have</td>
</tr>
<tr>
<td>Inability to pick up objects that are fastened (e.g. sink, stove, etc.)</td>
<td>Must have</td>
</tr>
<tr>
<td>Ability to experience different camera perspectives</td>
<td>Must have</td>
</tr>
<tr>
<td>Intractability (doors opening the right way, lights)</td>
<td>Must have</td>
</tr>
<tr>
<td>Ability to import custom objects</td>
<td>Must have</td>
</tr>
<tr>
<td>Ability to experience the environment from a different (human) perspectives (e.g. wheelchair)</td>
<td>Should have</td>
</tr>
<tr>
<td>Ability to measure distances between two points</td>
<td>Should have</td>
</tr>
<tr>
<td>Location of power plugs and internet plugs</td>
<td>Could have</td>
</tr>
<tr>
<td>Location to transfer cables (from the ceiling and the like)</td>
<td>Won’t have</td>
</tr>
<tr>
<td>Ability to alter furniture (size, material) in VR</td>
<td>Won’t have</td>
</tr>
</tbody>
</table>

4.2.1 Smooth Integration with SH3D:

Smooth integration with SH3D highlights the importance of being able to create the desired environment in SH3D and easily import it to Unity to use in VR. Allowing the user to explore different
environments of their creation. This requirement is also the core functionality of allowing researchers to plan and prepare the layout of the rooms to their needs.

Smooth Integration is used to describe the process that the user needs to undergo; minimal and non-burdensome. It should be no more than a couple of simple steps that anyone can undertake in order to get the setting working in VR. It should be noted that this is not in reference to the creation of the environment in SH3D, as that is not within the scope of this project.

4.2.2 Ability to Move Around:

The ability to move around is in reference to having one’s movement in reality transferred to virtual reality. An essential aspect of a virtual environment, as it conveys the feeling of presence, making the experience for users more engaging and immersive. Holding especially true for full-body movement (Slater, McCarthy, & Maringelli, 1998). Movement translation is one of the basic virtues of virtual reality and the premises upon which much of VR is built. The importance of movement translation has been known for over two decades.

The ability to move around will allow users to navigate the eHH as if they were there and enables features such as bending down to pick up objects. The movement translation will be done on a 1:1 scale; one meter of movement in reality becomes one meter of movement in the virtual setting. Tracking of the player will be done by recording the movement of three reference points; the head using an HMD, and the hands by held controllers.

The player will be tracked using the HMD, a feature native to the Oculus Rift S. Character movement is gathered from the position of one’s head, and so is the player’s location within the scene. To clarify, moving the headset one meter forward will cause the character position to move that much as well (which is not the case for moving the hands). The other two tracking locations are the controllers which the user will be holding in their hands. Allowing their hands, the main form of interacting with the environment, to be separately tracked from their head. Such a setup enables the ability to pick up objects as if one was physically there.

4.2.3 The secondary form of Movement:

The secondary form of movement refers to an alternate way of locomoting through the virtual eHH. Being able to do this is essential because if tracking is the only way to move throughout the environment, a space as large as the eHH would be required to use the application properly. Hence, another
A way of locomoting is required. As discussed in Section 2.3, locomotion is an essential factor, as it can cause cybersickness. In order to prevent this, the right form of locomotion is vital.

Where research shows that not all literature sees teleportation as the best form of secondary movement (Buttussi & Chittaro, 2019; Christou & Aristidou, 2017; Clifton & Palmisano, 2019; Coomer et al., 2018; Loup & Loup-Escande, 2019), it is the strongest overall contender from the list. Furthermore, despite them not all claiming that teleportation is the superior method, they do all acknowledge that teleportation seems to be the industry standard concerning locomotion. For this project, teleportation seems like the best method of locomotion.

4.2.4 Full Model of the eHH:

This requirement pertains to the model of the eHH itself. It has to encompass the meeting room, control room, and the living facilities that are found within the entirety of the eHH. In earlier stages of the project, the idea was only to model the living quarters, not the meeting room, nor much of the control room spare the mirrors.

4.2.5 Ability and Inability to Pick up the Right Objects:

The VR environment is built to specification and does not suffer from the same limitations that reality does, creating possibilities that would be impossible in the physical world. An example is, picking up the floor, rotating it, and throwing it away. The process of doing so would result in all the furniture, and user, falling into an endless void of nothingness. In order to prevent the loss of all the furniture and ensure the virtual eHH represents a realistic environment, it is vital to ensure that not all objects can be picked up. Including but not limited to, the floor, ceilings, walls, kitchen equipment (sink, cabinets), shower, sinks, toilets, or closets screwed into the wall. Further, objects such as pillows, chairs, couches, closets, beds and fridges should all be movable.

4.2.6 Ability to Experience Different Camera Perspectives:

The eHealth House has several cameras placed on the ceiling. These cameras can be used by researchers to record what is happening in the environment. Hence, it is vital that when an experiment is taking place, all the required angles are covered, and what needs to be seen can be seen by the cameras.

The virtual model should facilitate this feature by allowing researchers to experience the view of the different cameras from the virtual environment directly. By allowing researchers to do this, they will be able to consider the cameras when designing their layouts.
4.2.7 Intractability:

The environment should be made in such a way where it reflects reality. This requires a certain amount of interaction with the environment. Lights need to be able to be turned on and off, and doors need to be able to be opened and closed. Not only should doors be able to be opened, but they should also be opened in the same direction they would in the actual eHH. These are little details that will make the environment feel realistic. When the user is familiar with the physical eHH, the virtual one should feel familiar and correct. Such little details are essential in making that experience come into fruition. If one is familiar with the eHH and is in the virtual environment wanting to open a door, and it opens the other direction, it can be a jarring experience. Further, door opening directions are important with furniture design choices, including but not limited to, sizes and location of placement.

4.2.8 Ability to Add Custom Objects:

There is a high chance that a product will be used that is not available by default in SH3D, but the researchers should still be able to include such objects in their environment. Hence, it becomes vital to enable a feature that allows researchers to import custom objects into the environment. Custom object importation is a feature native to SH3D and should thus, not cause too many problems.

4.3 Specifications Sheet:

Create a 1:1 scale replica of the eHealth House, with as many details as possible.

Add virtual reality support to the environment, such that, a user can walk around the digital eHealth House. Preferably with the use of an Oculus Rift S headset.

Implement a feature that adjusts the user's height in virtual reality to their height in reality.

Implement a function allowing the user to teleport using the teleport button (reference controls). Allow them to teleport only to locations on the floor within the eHealth House.

Ensure that the doors have interactability where they can be opened and closed. Make sure that doors only open and close when the user grabs the door handle (using the grab button – reference controls) and moves that. Further, ensure that limits are in place so that the doors open as they would, in reality, and not the other direction.
Allow furniture that is movable, i.e. things that are not bolted nor stuck to a surface, to be picked up by the user using the grab button (reference controls). They should thus be able to move the furniture around.

Implement a physics engine. Objects that have gravity on them in real life, should have gravity on in the virtual environment as well. Releasing an object, such as a chair, should cause it to fall and interact with the environment around it.

There is an exception for paintings hanging on walls. They should not interact with gravity and should be kinematic. This way the user can ‘hang’ them anywhere on the wall.

No object should interact with the object that the user is holding, unless that object has been specified to do so. That is to say, given object A somewhere in the eHH, and object B that the player is holding, upon colliding between these two objects nothing should happen; unless A has been specifically instructed to interact with a held, in this case, object B.

The eHH has four cameras. Recreate these cameras in the virtual environment and allow the user to watch through the lenses of these cameras, by pressing the change camera button (reference controls). The user should cycle through these cameras, that every time the button is pressed, they go to the next camera. After the user saw through the last camera, go back to the user’s perspective.

The user should be able to easily integrate their Sweet Home 3D creation. They should only be required to specific to the program the location of their export. The program should then take the created environment, and automatically implement the required features. Likewise, objects that can be picked up, should be able to be picked up. Doors should retain their functionality.

Create an outside environment such that the user does not stare out over an endless void. Preferably, create an outside area that reflection the outside view of the eHH. Make this area inaccessible for the player through teleportation. However, ensure that it consists of a layer that user can still stand on, for if they decide to walk through a wall using physical movement.

Controls:

The controls are designed with an the Oculus Rift S headset and controllers.

The trigger is used to teleport. Pushing trigger creates a parabolic arc that is used to aim, where the user will teleport too. Releasing trigger should cause the player to teleport.
The grip button is used for picking up objects. Pushing the button grabs the piece of furniture that it is colliding with. Releasing the button releases the object. Allow no more than one object to be picked up by one hand at any one time.

The thumbsticks are used to rotate. Moving the thumbstick left rotates the user 45 degrees left. While moving the thumbstick right, rotate the user 45 degrees right.

The ‘A’ and ‘X’ buttons are used to interact with UI buttons. When ‘A’ or ‘X’ is pressed, press the UI button, provided that the controller who’s button was pressed was aiming at the UI button. When either of the buttons is touched, create a straight laser coming out of the controller, aligning with where the controller is aiming.

The ‘B’ and ‘Y’ buttons switch between camera angles when pressed.
5 Initial Approach:

While initially, a realization chapter can be expected after a specifications chapter, this is not the case. The paper aims to layout the process underwent in the development of the produced tool. It should be mentioned that it was not possible to finish the desired tool as described above; instead, an altered version of the tool has been produced. The foundation of the desired tool is to import an environment produced by SH3D into Unity and allow the researcher to use it with minimal to no interference. The altered tool can still do this to an extent, although, not as extensively as initially planned. The problem lies in the nature of how SH3D exports its files and the way Unity receives those files. Making the streamlined foundation envisioned between the two apps, not something that could be accomplished at the time.

This impasse was only discovered during the realization of the actual product and thus was not mentioned in sections 2 till 4. Further, simply starting to talk about what has been produced without a proper introduction is not feasible either. Starting to describe the development process of an altered tool without context as to why it is different or what is different would be very confusing; hence, the mention of it beforehand.

This section will first delve into the initial plan to create the desired tool, so the envisioned process can be seen. This will be followed up with an explanation of the two problems that prevented the desired tool from being created. The following section will be talking about what the actual tool has become and what it can do, providing the required context in order to understand how it was made. Once that has been done, it becomes possible to delve into the realization of the product.

One additional element that has to be specified is that from here on out, there will be a difference between the word’s object and object\(^1\) in italic. The difference is described in the footnote, as well as an explanation of parents and children in the context of Unity.

---

\(^1\) An object is what Unity considers one thing. An example, a single object chair would be a chair that is considered as one whole piece (i.e. it is a chair). A two-object chair might consist of the legs of the chair, and the seat of the chair (i.e. you do not have a chair, you have a chair seat, and chair legs). These objects are independent of each other. Further, it is possible for an object to be considered a parent or child. Objects are dependent on their parent object, but independent of their children. Changing an element such as a rotation, position, or size of a parent will likewise happen to children. However, making such an adjustment to the child will not affect the parent. An Empty GameObject is an object that has no presence in the virtual environment but functions the same as an object.
5.1 Planned Work Process:

This subsection explains what the initial plan was in developing the tool before the problems were identified. While a more in-depth explanation of the ideal work process could be included, since much of it is not relevant, for it was not possible to implement it, it has been excluded.

5.1.1 Sweet Home 3D:

SH3D would be used in order to model the eHealth House. Using different functions of the program, it becomes possible to model the walls, floors, windows and other interior.

5.1.2 Custom Modeling:

Any items and interior needed that SH3D could not provide would be custom modelled in Maya. This would then be imported to SH3D. This combination of SH3D and custom models would allow for a template of the eHH to be created. Said template would be a file that can be opened in SH3D and contains the elements that should not be changed about the eHH (shower, toilet, windows, doors, etc.).

The template functions as the base model of the eHH, what does it look like without furniture? All the user would have to do is open the template and drag and drop the furniture to where they want it. If they so desired, they could use more advanced features of the program to elevate the furniture, change its textures, rotate it, invert it, or change its size. Researchers would also be able to import custom furniture to the scene, in case their desired piece of furniture cannot be found, or if they made a product that they need to test.

5.1.3 Into Unity:

The user would only have to export the created environment and import it into Unity. Unity would be programmed in such a way to take the imported file and automatically add the required features to the required components.

The distinction between movable and non-movable:

One of the more advanced features of SH3D is selecting an item to be ‘Part of base plan’. Unity would know that anything which is ‘Part of the base plan’ should not be able to be interacted with. The template would have taken all the items that cannot be interacted with and marked them as ‘Part of base plan’.

Physics:
A script would be written to add physics to all pieces in the scene automatically. Accomplishing this can be done by adding something known as a ‘mesh collider’ and a ‘RigidBody’, giving the item physical shape and physics, respectively. Items that are ‘Part of base plan’ should be ‘isKinematic’ so that they will not move.

Grab-ability:

Another script, or the same one as above, would be used to automatically add any components to the pieces not marked as ‘part of base plan’. The added components would be the ones that are required in order for the player to interact with it.

Placement:

The created program would also automatically centre the model to the center of the world.

5.1.4 VR Player:

The developers who create Virtual Reality headsets, often create resources to be used on Unity in order to use their headsets. These resources will also be used in order to get a usable pre-configured player character. This preconfigured character would have the ability to teleport and grab furniture. If not, this would be added manually. The character would be placed in such a way, that when the imported model is centred in the world, the player is in the centre of the control room.

5.1.5 Doors:

Doors are a bit more intricate than regular pieces of furniture. Not only would they require the physics of usual furniture, but they would also require a hinge joint, a component that sets an axis of rotation at a specific point. The hinge joint could thus be used to allow the door to only rotate around a particular axis at a certain point, for a certain number of degrees in each direction. The hinge joint would automatically be added to the doors.

Cameras:

One crucial element of the product as to be able to see from different perspectives, namely the CCTV cameras in the eHH. These would be separately modelled in Unity and manually placed. This way, once the user opens the created application, the cameras are already there. They are placed in such a way, that when the user imports a scene from SH3D, and the program centres the environment, the cameras are in the right location and ready to use.
5.1.6 Synopsis:

The above paragraphs briefly state the main ideas of how the development of the tool was initially planned. While quite some detail is missing on how everything was to be implemented, it was not relevant to include, for this section is only supposed to give an idea on how the ideal could be done. Some aspects of the realization reflect what happened here and will be explained more in-depth there. The next subsection is dedicated to explaining the two main problems that were faced, which made the ideal implementation impossible.

5.2 The Problem:

The reason there were discrepancies between the initial idea and the developed tool can be narrowed down to two main reasons. The way SH3D exports its environments and the way mesh colliders work.

5.2.1 Exporting of the eHealth House:

SH3D exports its model by taking the name of the object and adding a number to it. The number was simply a reference to when the object was exported. The first object to be exported will be ‘AAA_1’ and the second will be ‘AAA_2’. Where ‘AAA’ is the name of that object. The exported objects do export in order. That is to say, a chair consisting of three parts will have the numbers ‘X’, ‘X+1’ and ‘X+2’.

The way an object is named is determined by who made the object. Furniture in SH3D is constructed by different individuals, all of whom use different naming conventions. Some creators would give their objects proper names. When designing a table, the objects that make up that table might use the name ‘TableLegs’, or ‘TableTop’. Other creators use less meaningful names like ‘1’ and ‘2’ or keep the default of the original object from which the pieces were made; names such as ‘box’, ‘cube’ or ‘polymesh’. It is also possible that particular objects have no name, and when exported will simply get a number as their name.

At first, the problem arising from this might seem trivial or non-existent. However, the naming convention used by exporting thus makes it impossible to tell what objects belong together in any way, shape or form. Further, the numbering convention used does not allow for furniture pieces to be grouped easily together. While it is possible to know that objects belonging together have numbers that are close to one another, it is not possible to know when they switch from one piece of furniture to the other. With no way to extrapolate this, it becomes impossible for a script to determine which objects need to be grouped
together; resulting in the only way it can be done, by human input. The script cannot know that an object by the name “231” is the seat belonging to chair legs “235”. Only when looking at it, and manually selecting the pieces, does this become apparent. This means that the ‘physics and grab-ability’ and ‘doors’ from section 5.1.3 and 5.1.5 cannot be automated.

5.2.2 Mesh Collider Inaccuracy:

Colliders are needed in order to identify the bounds of an object in such a way that Unity understands it. The collider determines where the object starts and ends. Without a collider, also known as hitbox, the object does not take up physical space. Where different colliders exist, such as a box collider (which places a box around the object), a sphere collider (which places a sphere around the object), so does a mesh collider. A mesh collider bases the shape of an object on the mesh, the textures of an object, to determine its boundaries.

While sometimes mesh colliders will correctly model a piece of furniture, so no additional work has to be done, this is not always the case. This is determined by the complexity of the object’s shape, as illustrated in Figure 5.

![Figure 5: A table with mesh collider (green lines)](image)

The table in Figure 5 consists of two individual objects. One object contains the four legs and top of the table (all brown pieces), while the second object is the grey piece connecting them. As can be seen,
each of the two objects is surrounded by green lines. These green lines represent the mesh collider. For the object consisting of the top and legs of the table, however, the entirety of the space below the table is also included; thus, according to Unity, the entire space below the table is solid and takes up space.

As seen in Figure 5, the mesh collider for the object consisting of the top and legs of the table also includes the space below the table, making the space below the table occupied. Which means that placing a chair under the table will result in placing two (or more) objects inside of each other, and when the program is run, they will rapidly be pushed away from one another – as two objects cannot occupy the same space. While this is possible to change by altering the layer that the chairs are on, then the table and chairs will no longer be able to interact with each other. Not doing anything and just placing the furniture where it belongs will result in the table and chairs to be launched across the room, potentially clipping out of the eHH, upon starting the program. Further, as they are launched away, they will likely collide with other furniture, causing those to fall over or be launched across the room as well.

For particular furniture, namely tables and chairs, these inaccurate hitboxes happened very often, resulting in very unrealistic scenarios. This is very well illustrated in Figure 6.

Figure 6: Interaction between a pillow and an armchair both of which have a mesh collider
This is how users would see and interact with the virtual world around them. Pillows that are floating above chairs and couches, and the inability to place chairs underneath tables, which means that for any piece of furniture that had this problem, custom models which approximate the bounds of that piece of furniture, had to be made.

5.2.3 Implications of the Problem:

The two main problems discovered prevented the creation of a streamlined process between SH3D and Unity. The resulted product is thus, guaranteed to be different then what was initially intended. This will be further explored in the next section.
6  Actual Product:

As mentioned in the previous section, to ensure that the realization section is clear, first, the resultant product will be explained. The produced tool, namely due to the problems discussed earlier, is not a streamlined process between SH3D and Unity. It is, however, still possible to import one’s custom environments.

The produced tool provides the user with a tutorial and four custom environments to explore. These are shown in Figure 9, Figure 10, Figure 11, Figure 12, and Figure 13.

Controls:

Since the user is in virtual reality, using an Oculus Rift S, they are able to move their head and walk-in reality in order to have that movement translated into VR. However, they are still tethered to a computer and likely have limited space. In order to navigate and interact with the scenes, users are provided with controllers. Both the controllers and their button mapping (what button does what) can be seen in Figure 7. One can use their index finger to teleport, and their middle finger to grab. The joysticks can be used to rotate oneself in increments of 45 degrees at a time. The ‘A’ and ‘X’ buttons are used to interact with UI elements, and ‘B’ and ‘Y’ are used to cycle through the different camera perspectives.

Tutorial:

When the application is run, users will be placed inside the control room of the tutorial scene. For reference, look at Figure 9. The user is instructed on how to rotate and teleport and is then asked to teleport to the green squares (only one is visible at a time when the application is running). After they teleported a couple of times, they will be asked to pick up objects and place them on a pedestal of matching colour; this can be seen in the top right Figure 9. After this, they will be asked to open a door and cycle through the different cameras. The tutorial is finalized by pressing a UI element (button) which takes them to scene 2.

Scenes:

There are four scenes that users can explore. The first scene (Figure 10) is the eHealth House when empty, only including ‘furniture’ that cannot be moved. This allows the user to see what the eHH might look like when empty. It is also possible to import one's custom environment to the empty scene, although, this setting will not be interactive. A guide on how to add a custom setting can be found in Appendix 2 –
Advanced User Guide – How to Import Custom Furniture: It should be noted, importing one’s own setting is slightly more advanced.

The second scene (Figure 11) contains some furniture that the user can pick up and move around. This allows them to see what the eHH might look like if the furniture gets added. Scene three (Figure 12), is made to build off of scene two. In scene three, much of the furniture is the same, plus a whole lot of additional furniture. Some extra lamps, a teddy bear, guitar, wheelchair, bread, and a coffee machine, to name a few. Scene two shows what a researcher might design a setting like, while scene three shows how it could be when someone is living in the environment.

The fourth (Figure 13) and the final scene is somewhat different. The furniture it contains is purposely thematic, made of the same material, in similar styles. This scene not only highlights the variety of settings that can be found within the eHH but also shows one of the substantial advantages of the tool. The grand piano, which is found in this scene, is a testament to the ability to add and design rooms for oneself at a fraction of the time it would take to do so in reality. Adding a grand piano only took a few minutes, however, doing so in real life would be an expensive, lengthy ordeal. Functioning as a reminder to the user, what benefits such a tool can bring.

Camera Perspectives:

The user can experience different camera perspectives. The eHealth House has multiple cameras that record everything that happens. In order to allow researchers to see what the provided settings, or their own created ones, might look like from these cameras, the user can opt to experience the virtual environment from the perspectives of the cameras. An example of this can be found in Figure 8.

Outside World:

In an attempt to make the VR tool more immersive, all the scenes also have a view to the outside world. While this view is not an accurate representation of the view one would have in reality, it does result in a more immersive environment then simply looking at an endless void. This outside world consists of a small walkway around the eHealth House (which is there in reality as well) and a grass field. The grass field is surrounded by hedges, providing a border to the endless void. Trees have been placed both on the grass field and behind the hedges. Most of this can be seen in Figure 14.
Figure 7: User Controls – Original picture by Oculus (2020)
Figure 8: A screenshot of one of the camera perspectives, allowing the user to see the eHealth House as the cameras do.
Figure 9: A top-down view of the eHealth House tutorial, teaching users how to use the controls – Scene 0
Figure 10: A top-down view of the Empty eHealth House scene created. Can also be used to import custom settings – Scene 1
Figure 11: A top-down view of the eHealth House with a moderate amount of furniture. Also, an example of what a researcher might plan their setting like – Scene 2
Figure 12: A top-down view of the eHealth House containing more furniture. Also an example of what the eHealth House might look like inhabited – Scene 3
Figure 13: A top-down view of the eHealth House containing a grand piano. Also, functions as a reminder of the usability and advantages of the tool – Scene 4
Figure 14: A top-down view of both the empty scene and the outside world that can be seen from the windows
7 Realization:

Now that the final product has been established, it becomes possible to delve into how it was made. This section is dedicated to that. Due to the production of the tool being an iterative process and differences existing between different versions of the program, this section will mainly talk about the latest version. It should be noted that other versions existed and have been improved upon. Notably, the first prototype, which has been tested during an interview with Wezel (2019). This provided several useful insights into how the tool could be improved.

7.1 Production:

The following subsection will delve into the production process for the creation of the tool.

7.1.1 Sweet Home 3D:

The premium version of SH3D was purchased. The free version comes with 100 pieces of furniture and 26 textures, while the premium version has 1400 pieces of furniture and 418 textures. This notable increase allows for more accurate modelling of the eHH, especially when elements such as windows come into play.

The client provided a copy of the blueprints of the eHH. SH3D allows for the importation of images to be used as a reference. For this reference image, a scale can be set that correlates the distance of the image to size in reality. The modelled environment resulted in an accurate representation of the eHH. The surface area of the modelled eHH is only about 0.82% larger than it is in reality. This was calculated by taking the surface area of the model (110.5m$^2$) and dividing it by the actual surface area on the blueprints of (109.6m$^2$). The reason for this inaccuracy is likely because the wall thickness in the model was not adjusted to be reflective of the wall thickness of the eHH. So the walls in the model are slightly thinner than they should be.

The height of the eHH (3m) was also measured in order to ensure that it was accurately portrayed in the model. The control room has an irregular ceiling of varying heights. The lowest of these points was used for the digital model at the height of 2.15m. The bathroom has a ceiling height of 2.8m, which is reflected in the model as well.

7.1.2 Into Unity:

Exporting the SH3D file to OBJ format allows it to be imported into Unity. When exporting, it was ensured that it was exported to its own folder. Dragging the entire folder into the asset manager of Unity
imports the OBJ. Opening the folder in the editor and dragging the prefab into the scene will create the entire eHH as designed by SH3D in Unity. The position had to be manually set to the centre of the world (0,0,0), and the scale had to be adjusted to (1/100th) of what it was.

The floors and ceilings created in SH3D were deleted and replaced with plane objects in Unity. This is because the floors from SH3D are nonexistent under the walls resulting in gaps. These gaps would be more prone to bugs; thus, custom floors were added. Further, the stairs and floor of the control room were modelled in Unity using cubes. All floor objects got their own layer.

7.1.3 Custom Modeling:

Not everything could be adequately created using Unity or SH3D; in order to solve that Maya was used. This allowed for items to be custom made to represent the eHH accurately. The kitchen and control room table (the one attached to the wall) were both created in Maya. These were then exported to Unity and imported into the scene.

7.1.4 Grouping:

Grouping is when one or multiple objects are taken and are made a child of another object. This way, by moving the parent object, the children also move. It also makes it possible to organize the layout of all objects on a scene. The way grouping was done was by pressing “CTRL” + “G”. This made all the selected objects a child of a newly created Empty GameObject. This was made possible by using a script provided by bjennings76 (2015).

Each piece of furniture was individually and manually grouped. After the grouping had been completed, the new group of objects got moved up. This was done to see if any objects were excluded in the grouping process. For the excluded objects would remain in place as they are not parented. The moved group was placed back in its original position by undoing the last action, and if any object was missed, it was added to the group.

Different groups were made and combined in order to produce a structure — several categories were made such as ‘Players’, ‘UI’, and ‘CameraManager’.

7.1.5 Adding Physics:

In Unity, two main elements are needed to add physics to an object. Colliders and RigidBodies. Colliders are what determine the bounds of an object, where it starts and where it ends. Whereas
RigidBodies give it mass and allow gravity to interact with the objects. Because of this, the adding physics subsection is divided into two parts.

7.1.5.1 Adding Colliders:

Any object that needed its bounds declared, ranging from windowpanes and walls to the small fitting on the bottom of the feet of a chair, where given mesh colliders. At times mesh colliders do not function, as explained in section 5.2.2, colliders were roughly modelled using box colliders. This can be done by taking a Cube GameObject and resizing and shaping it to the desired shape. When multiple cubes are combined together, they can create a rough outline of a piece of furniture. The boxes had their mesh turned off, meaning that they were invisible, but still took up physical space. This allowed pillows to be correctly laid on couches and chairs, as well as chairs to be placed under tables; as can be seen in Figure 15 and Figure 16, respectively. Creating custom colliders were more accurate than mesh colliders and allowed the program to run correctly.

Figure 15: A armchair and pillow with custom hitbox colliders (green lines)
Other solutions were attempted to no avail. Two plugins, RASCAL Skinned Mesh Collider by Bolt-Scripts (2019) and Non-Convex Mesh Collider by productivity-boost.com (2017), both claim to solve the issue by mesh colliders, but neither did.

7.1.5.2 Adding RigidBodies:

RigidBodies tell Unity to treat this GameObject and its children as a single physical piece. If a chair consists of three pieces, the pieces were made a child of an Empty GameObject. Provided that mesh colliders worked for the piece of furniture, mesh colliders were added to all children. Then a RigidBody was added to the Empty GameObject parent. This will cause all the bounds of the object to function in unison and be treated as the bounds of a single item.

Some pieces do not need to interact with gravity, and others should not interact with anything. To change these features, one can check or uncheck the ‘gravity’ or ‘isKinematic’ booleans, respectively.

7.1.6 Adding grab-ability:

For the objects that need to be able to be picked up, the ‘OVR Grabbable’ script was manually added. This would be done to the same object/parent that had the RigidBody. Further, the ‘OVR Grabbable’ script is altered in such a way that it functions with the hitboxes of its children automatically. Under normal
circumstances, the user would have to input which hitboxes work manually, if it is referencing hitboxes of its children. This would have to be done for each and every piece of furniture. In order to help with the ‘OVR Grabbable’ implementation, several tutorials were referenced (Valem, 2019b, 2019c).

Lastly, all the furniture pieces were placed on the “GrabbedObjects” layer. Certain smaller objects were placed on the layer “GrabObjects”, the only difference between the two is, that “GrabObjects” interact with items that the user is holding, while “GrabbedObjects” does not. This allows the program to run in a more functional way to the user, preventing issues arising by moving furniture. As an example, it becomes possible that when a user picks up a couch, the pillows on it will stay on the couch, however, the couch will not move other furniture around the room such as tables, chairs or lamps.

7.1.7 VR Player:

The implementation of a VR player ended up being a bit more intensive than initially thought. In order to enable the use of VR, Oculus Integration by Oculus (2019) was used. These are scripts, functionalities and prefabs created by the developers of the headset to be used in Unity. In order to implement the plugin and use a VR character, multiple tutorials were used (Valem, 2019b, 2019c, 2019e). Merely adding the ‘OVR Player Controller’ prefab enabled a basic VR experience. Movement translation from reality into VR works flawlessly.

The ‘OVR Player Controller’ also automatically causes the player to move with the use of the controller’s joystick. This feature has been disabled as it caused cybersickness, an element brought up in Wezel (2019).

7.1.7.1 Teleportation:

Since joystick movement has been removed, teleportation was added. This was done according to the tutorial Valem (2019e). An Empty GameObject was added to the ‘OVR player Controller’, named ‘Teleportation’. To this the scripts: ‘Locomotion Controller’, ‘Locomotion Teleport’, ‘Teleportation Input Handler Avatar Touch’, ‘Teleport Target Handler Physical’, ‘Teleport Aim Visual Laser’, ‘Teleport Orientation Handler Thumbstick’, ‘Teleportation Aim Handler Parabolic’, and ‘Teleportation Transition Blink’ were added. Instead of describing what settings were changed, they have been depicted in Figure 17.

Then the prefab for TeleportDestination was added to the scene. When teleporting, a giant compass rose is created to see the teleport destination. In order to change that, this prefab was altered as described in Valem (2019e). Additionally, the arrow in the centre of the prefab had also been removed.
Settings were altered in such a manner that the user teleports to the destination they indicated and ended up facing in the same direction they were before teleportation, this is based on observations during the interview with Wezel (2019).

The teleportation did have some bugs, which has been solved by the use of the script provided by Debashishb06 (2013).
Figure 17: Teleportation Settings
7.1.7.2 Hands:

Merely adding the OVR Player Controller allowed for the controllers to be used to a limited degree. This in of itself did not allow objects to be interacted with, or hands nor the controller to be seen inside the virtual environment. One would be swinging their arms around, and none of that would be visible in VR. This was solved by adding a ‘LocalAvatar’ prefab as the child of the ‘TrackingSpace’ inside of the OVR Player Controller. Moreover, as children to the ‘LocalAvatar’, the ‘AvatarGrabberRight’ and ‘AvatarGrabberLeft’ prefabs were added. The full final OVR Player Controller can be seen in Figure 18. The resultant hands allowed for not only hands to be seen, but also the controller that the player is using. This makes the use of controllers easier, as the player can see what they are pressing. Touching a button also moved the player’s respective finger in VR to the button being touched. This way, the user can see what they are doing on the controller without actually being able to see the controller nor their hands in reality.

Figure 18: OVR Player Controller and its children
7.1.8 Interactive Doors:

Creating functional doors was a lengthy process. It was done by the use of the tutorial by Valem (2019a). It involved adding a hinge joint to the door and altering the settings, so it moved correctly. As well as adding several cubes that interact in a specific way with each other, the door, and the player. An extensive long process to get properly working in the scene, due to lots of bugs and complications. For an in-depth explanation on how to do it, reference Valem (2019a).

7.1.9 Adding Cameras:

Different cameras were placed in the locations around eHH. The way Unity handles cameras is that the latest activated will be the one used. In order to alter the user’s camera perspective, the desired camera had to be activated. Then when the user wants to switch back to the first-person view, deactivate the camera.

In total, there were five different camera perspectives, one first person perspective, and four placed cameras; placed corresponding to their location in the real eHH. The way that another camera could be selected was by pressing the ‘B’ or ‘Y’ button, right and left hand respectively. Pressing one of these buttons would allow for cycling between: “Player -> CeilingCamera1 -> CeilingCamera2 -> CeilingCamera3 -> CeilingCamera4 -> Player” after this the cycle would repeat.

A script was written to change the tracking point once a camera gets switched. When the camera is from the user’s perspective, it will be tracking from the Floor Level; meaning the height of the player’s perspective is dependent on the distance between the user and the floor. While if it is from one of the ceiling cameras, it will be done from eye level, the location at which the cameras are placed.

7.1.10 Adding Tutorial:

A tutorial scene has been added in order to teach the user how to run the program. The scene is illustrated in Error! Reference source not found.. It was created by placing several pads to which the user needs to teleport. They have to solve a small puzzle, as seen in Figure 19. The object on the white table has to be placed on the pedestal of the respective colour, before the user can continue. This was done by adding a function in the script that only allowed the next teleportation pad to become visible when a certain Boolean was activated; done by solving the puzzle. The user is then asked to open a door, cycle through the CCTV cameras, and lastly, press a UI element to finish the tutorial.
In the prototype tool, the tutorial was a completely different scene irrelevant to the eHH. While it did help in teaching with controls, it caused some problems, some text was difficult to read, and some tasks seemed a bit too vague. This was observed during the interview with Wezel (2019). This is another reason as to why the tutorial scene was remade entirely.

7.1.11 Adding a User Interface:

The way the UI had been implemented inside the eHH, is by placing text on the wall of the control room, as seen in Figure 20. This UI consisted of a welcome message, a small explanation of the controls, and five UI elements buttons, made to change the scenes. The UI was based on the tutorial by Valem (2019d).
In an attempt to keep things as minimalistic as possible, only the required information has been included on the UI. Additionally, a feature that has been added is the addition of a line that shoots out from the controller when the ‘interact with UI element button’(‘A’ or ‘X’)’ is touched — allowing users to aim more clearly. Something that became apparent with the prototype testing was that pressing UI elements was quite tricky as it was difficult to aim.

7.1.12 Adding Different Scenes:

Different scenes were implemented by taking the empty scene, duplicating it and importing furniture from SH3D. Furniture that worked with mesh colliders was immediately placed in the environment, while furniture that required custom colliders had their colliders made and placed when functional. By the use of this method, three different scenarios were created.

The interview after the prototype provided insight into the usefulness of scenes and furniture in the eHealth House. The addition of the piano scene hopefully allows the user to see the environment from, yet another perspective as well interacting in a way that was not possible in the other environments.
7.1.13 Adding Outdoor:

After the scenes were created, an outdoor environment was designed as well. This was to make the virtual eHH more immersive and enjoyable for the user. It would also mean that the user is not looking out on towards an endless void, but instead gets to see a garden. This is also the reason that the garden has an incline; ensuring that the user would see the garden, and not beyond it. Further, a hedge has been placed at the edge of the garden, functioning as a boundary between the endless void and the user. Trees have been placed both in the garden and beyond the boundary to make it feel like the world does not end with the hedge. A view from the eHealth House into the garden can be seen in Figure 21.

Figure 21: A view of the garden from the eHealth House

7.1.14 Miscellaneous and Synopsis:

Another difference between an iterated design and the final version was the button mapping. Button mapping has been changed so that they are identical across two controllers. This means that the application can be run with either one controller or both at the same time while retaining all its functionality. Users no longer need to worry about which controller is used for what, but only must lean the button layout for one controller. Prior to this, both controllers were needed to explore the environment, and both were needed to use all the functionality that the tool had to offer.
In all scenes that had lamps, point lights were added. This way, the lamps produce light and make the environment seem lively. For each scene, directional lighting has been added as well to ensure there is enough light and that things remain visible.

This section provided an insight into how different aspects of the tool were produced and some of the design choices that went into this process. Allowing one to not only understand what the product was but also how it functions and what elements and components were used.
8 Evaluation:

This section is dedicated to assessing the tool's successfulness. Assessment will be done based on three metrics. Comparing the tool to the requirements established in section 4.2, comparing the tool to insights gathered from user testing, and exhibiting the tool at a demo. This enables the tool to be evaluated in a fair manner and provide insight to not only what is functional but also what was subpar.

8.1 Comparing the Tool to the Requirements:

In order to allow the proper comparison of the tool to the requirements, a copy of the table has been added for easy reference. To this table, an additional column is added, indicating if the requirement was: Implemented (requirement met), Limited (requirement reached to a certain degree), Not Implemented (requirement not met). This can be seen in Table 2.

*Table 2: Table containing requirements, priority and a column indicating if the requirement has been met*

<table>
<thead>
<tr>
<th>Feature:</th>
<th>Priority:</th>
<th>Accomplished:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Smooth Integration with SH3D</td>
<td>Must have</td>
<td>Limited</td>
</tr>
<tr>
<td>Ability to move around</td>
<td>Must have</td>
<td>Implemented</td>
</tr>
<tr>
<td>A secondary form of movement</td>
<td>Must have</td>
<td>Implemented</td>
</tr>
<tr>
<td>A full model of the eHH</td>
<td>Must have</td>
<td>Implemented</td>
</tr>
<tr>
<td>Ability to pick up objects (that can be picked up in real life)</td>
<td>Must have</td>
<td>Implemented</td>
</tr>
<tr>
<td>Inability to pick up objects that are fastened (e.g. sink, stove, etc.)</td>
<td>Must have</td>
<td>Implemented *</td>
</tr>
<tr>
<td>Ability to experience different camera perspectives</td>
<td>Must have</td>
<td>Implemented</td>
</tr>
<tr>
<td>Intractability (doors opening the right way, lights)</td>
<td>Must have</td>
<td>Limited</td>
</tr>
<tr>
<td>Ability to Import Custom Objects</td>
<td>Must have</td>
<td>Implemented*</td>
</tr>
<tr>
<td>Ability to experience the environment from a different (human) perspectives (e.g. wheelchair)</td>
<td>Should have</td>
<td>Not Implemented</td>
</tr>
<tr>
<td>Ability to measure distances between two points</td>
<td>Should have</td>
<td>Not Implemented</td>
</tr>
<tr>
<td>-----------------------------------------------</td>
<td>-------------</td>
<td>----------------</td>
</tr>
<tr>
<td>Location of power plugs and internet plugs</td>
<td>Could have</td>
<td>Not Implemented</td>
</tr>
<tr>
<td>Location to transfer cables (from the ceiling and the like)</td>
<td>Won’t have</td>
<td>Not Implemented</td>
</tr>
</tbody>
</table>

*Not available in a custom environments*

Requirements that have been ‘Implemented’ will not get explored in-depth. The requirements have already been explained prior, and if ‘Implemented’ have been met. All the elements with ‘Not Implemented’ will not be explored at all. None of these was a ‘Must Have’ for the product, indicating that it was not required to function. The reason the ‘Not Implemented’ were not included was due to time limitations. Two items are ‘Limited’ though and will be explored in further depth. Before these two requirements are looked at, a quick look will be taken at the ‘Implemented’ features.

The HMD tracks movement, and the user can also teleport using the controllers as a secondary form of movement. The models of the eHH are modelled in full and are accurately made to a 1:1 scale. The user can pick up objects that should be movable, while ones that should not be movable cannot be picked up. The user can experience eHH from different perspectives, including the control room and cameras. They can also import their custom objects to create custom environments through SH3D.

Intractability is a requirement that received ‘Limited’; the user can fully interact with the doors and said doors also only open in the direction that they should. However, they can only move when the user is grabbing the handle. Thus, pushing a door that is ajar, or even grabbing the door and pushing it will have no effect. The user is required to grab the handle. Intractability with light, being able to turn lights on and off, has not been integrated whatsoever and is thus lacking. The lamps are always on. One scene adds intractability by including a table tennis racket and a bucket of balls the user can interact with.

Regarding the smooth integration with SH3D, the user is still able to import custom environments; however, doing so is slightly more of a hassle than initially intended. In a custom environment, most requirements are still met. The user would be able to see their imported environment from all perspectives, and it would include the entirety of the eHH. The ability for movement, both physical and secondary, would still be intact. Intractability would still have ‘Limited’ functionality; however, none of the lamps would be producing light. The user will still be able to interact with the doors as they would in other scenes. Items with the asterisk will be different in a custom environment. None of the objects that should be movable
can be picked up, which would include any custom object. On the bright side, the ones that should not be able to be picked up will not be either.

8.2 User Testing:
This section will delve into the user testing done in order to see how effective the tool was.

8.2.1 Introduction to User Testing:
In order to test the use of the VR tool, user testing was conducted. Individuals were requested to complete several tasks and fill out a questionnaire based on their experience. The questionnaire gathered both qualitative and quantitative data. Things of note that became clear when performing the user testing for an observer have been noted down, and will also be used in the analysis.

8.2.2 The Candidates:
Seven people (n=7) have done user testing (1 female). Of the testers, the average age was 20.1, with a standard deviation of 1.46 years. Out of all the candidates, only two have seen or been in the eHH. One user has never used VR before, four users have limited experience (pertaining to once or a couple of times), one user has a moderate amount of experience (using it every now and then), and one user is very familiar (or using it often, or having experience in making VR applications).

8.2.3 Testing Setup and Protocol:
Participants conducted the testing in a room where the approximate area they had to walk around was 1.3x1.3 meters. At times, extending one’s arms was possible beyond the boundary.

The questionnaire was made in such a way where both quantitative and qualitative data could be gathered. A Likert Scale was used, with values from 1 (lowest) to 7 (highest). Most Likert scale questions pertain to the ease of use, while to get opinions on the pros and cons of the app, users were requested to provide written answers. Things of note that became clear when performing the user testing for an observer have been noted down, and will also be used in the analysis.

The user testing is divided up in 4 stages, as follows. For a broader breakdown on the questions asked in each stage, refer to Appendix 3 – User Testing Results:

---

2 An example question being: “I believe the tutorial was intuitive?”, participants were able to answer a whole number value from 1 (not intuitive at all) to 7 (very intuitive) Another example question: “How easy was it to pick up and move furniture?”, participants were able to answer a whole number value from 1 (very difficult) to 7 (very easy)
Stage 1:

1. Participants come in and fill out a form, in which they agree to participate as a user tester.
2. Participants answer questions about themselves, experience with VR and knowledge about eHH.

Stage 2:

1. Participants do the tutorial (the tutorial that is included with the tool see section 6)
2. Participants answer questions about the tutorial.

Stage 3:

1. The participant is asked to do the following tasks:
   a. Open "Lots of Furniture" and place the guitar and wheelchair on the bed. Then look at this via the bedroom CCTV camera.
   b. In "Some Furniture" go to the meeting room and place three chairs on the table there. Then look at it from the control room.
   c. In any scene, open all the doors
   d. In the "Piano" scene, take the grandfather clock from the living room and place it in the bedroom. Take the desk with the mirror (next to the door) from the bedroom and place it somewhere nice in the living room.
2. Participant answer questions about tasks

END TESTING

Stage 4:

Stage four was not planned initially. Nevertheless, due to the unclarity of some of the questions, participants were asked to answer new, more-clarified questions. These questions were asked post hoc. This is further explained in Appendix 3 – User Testing Results.

Further explanation:

The tasks were designed in such a way that the participant will have to use all the features in the eHH. The participants will need to open at least three scenes; thus, they get to practice using the user interface.
For three of the tasks, they have to move furniture. Due to this, the participant gets to experience moving furniture within rooms and to other rooms. By having to place furniture on top of other furniture, they are taught more about the physics system and how it interacts (an example being, placing two pieces of furniture inside each other results with the furniture being launched across the room).

For one task, they are required to open all the doors; thus, they get plenty of experience with those as well.

For another task, the user is required to cycle through the CCTV cameras. In hindsight, this could have been included in a second task.

Also, for one task, the individual is asked to look at something through the control room mirrors. This highlights that the control room also functions to let researchers observe.

8.2.4 Analysis:

The analysis is going to happen on a thematic basis. As an example, everything pertaining to the tutorial will be covered in one go, and everything pertaining to another function/aspect of the application will be covered separately. This allows users’ responses to the Likert questions, writing questions and notes to be integrated fluidly on a topic per topic basis. The results of the user testing are found in Table 3.
**Table 3: Table of Results of the User Testing, containing Question, Average Score (out of 7), and Standard Deviation**

<table>
<thead>
<tr>
<th>Questions Regarding Tutorial:</th>
<th>Average:</th>
<th>SD:</th>
</tr>
</thead>
<tbody>
<tr>
<td>I believe the tutorial was intuitive?</td>
<td>5.29</td>
<td>1.38</td>
</tr>
<tr>
<td>I believe the tutorial was beneficial in helping to understand what I can do?</td>
<td>6.14</td>
<td>0.69</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Questions Regarding Tasks:</th>
<th>Average ((\bar{x})):</th>
<th>SD ((\sigma)):</th>
</tr>
</thead>
<tbody>
<tr>
<td>How easy was it to pick up and move furniture?</td>
<td>6.29</td>
<td>0.76</td>
</tr>
<tr>
<td>How easy was it to get around the eHealth House?</td>
<td>5.86</td>
<td>0.90</td>
</tr>
<tr>
<td>How easy was it to open and close doors?</td>
<td>4.29</td>
<td>1.11</td>
</tr>
<tr>
<td>How easy was it to interact with the User Interface?</td>
<td>6.14</td>
<td>0.38</td>
</tr>
<tr>
<td>How easy was it to interact with the controller?</td>
<td>5.29</td>
<td>1.11</td>
</tr>
<tr>
<td>How easy was it to cycle through different perspectives?</td>
<td>5.86</td>
<td>1.46</td>
</tr>
<tr>
<td>How was your experience cycling through different perspectives?</td>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td>If you were to go to the eHealth House, how well of an overview of the environment do you think you'll have?</td>
<td>6.43</td>
<td>0.53</td>
</tr>
</tbody>
</table>
8.2.4.1 Tutorial:
Participants believed that the tutorial was beneficial in helping them understand what they can do in this environment. On average it got a 6.14, out of 7 (and henceforth, all scores of an average will be out of 7 unless mentioned otherwise), with a standard deviation of 0.69. This indicates that the results were mostly centred around that value, with some deviation.

People generally did like the tutorial. One user mentioned: “linear path of the tutorial makes it impossible to get lost spacially-wise” while another participant mentioned “The VR environment was convincing, interacting with the objects within it was intuitive. The door physics and such worked exactly as expected without having to be told.” User feedback also indicates that “If you followed it neatly, it could be done really quickly, though still guaranteeing that the user experienced the things that they can do.”

However, the tutorial is not without flaws. The placement of some instructions threw some people off. At one point in the tutorial, the participant is asked to open a door. The instruction on how to do this are located above the door they have to open, a door that is is about 2.4m tall. One tester commented “Maybe the orientation of some signs. E.g. I do not look up a lot and because of that found myself lost for a moment.” Some participants tried opening the door before they should, as they pass the door earlier in the tutorial. “Perhaps have the sign above the door pop up after the color sorting.”

Further, phrasing could be improved as well. One user mentioned “I had to read the text a few times to understand what buttons had what functions”, as well as, “Spell check (bellow)”. Additionally, in the section in which users were required to match the colours, one user tried to grab the pedestals.

8.2.4.2 Picking Up and Moving Furniture:
Participants can pick up furniture and move around with it. In regard to ease of use, testers found this the easiest, with an average score of 6.29. The standard deviation being 0.76 indicates that there is some deviation from that score.

It should be noted that this question can be somewhat lacking in information. It does not give proper justice to the physics engine on which the game is run. While one tester commented “Loved the physics” the same tester, when asked about the weakest aspects of the program, replied, “also the physics”. Another participant said, “Physics still were a bit glitchy, spassing out if you place objects within each other.” When two items are placed inside each other, Unity will attempt to ‘fix’ that as quickly as possible, resulting in both pieces to be propelled across the scene. Despite this, users did appreciate the fact that they were able to move furniture without exerting the same effort they would have to in reality. “Plus, it
adds a sense of comfort in not HAVING to do any effort to do the tasks that in real life would be a difficult and heavy chore.”

8.2.4.3 Getting Around:

On average testers scored getting around at a 5.86 with a standard deviation of 0.90. This indicates that people deviated from this average value. Regardless, the score shows that overall, people seem satisfied with the ease of use.

It should be noted that participants can teleport through walls and doors, making it possible to near-instantly move across the entirety of the eHealth House. One user commented, “It is more comfortable to phase around the place like a magical entity than it would ever be to just see it in real life.” Another user mentioned, “After you get the hang of it, there are a lot of possibilities of moving around and designing a place.”

Not everyone agreed on the positivity of movement possibilities. One user wrote, in response to what they believed the weakest aspects were, “the fact that you can only move "binary" no movement except point-and-click allowed”. Although, the tutorial does state moving around is a possibility by physically moving around, it is possible this was either phrased poorly, or the participant missed it. The tester no doubt, experienced some translation of real-life movement to VR, it appears to have been smooth enough to feel natural to them. Another tester wrote: “the fact that you are able to teleport through walls. That is not necessarily a bad thing, but (in this case) makes it a bit more confusing to where you go and where you came from.” Similarly, another participant wrote: “The ability to look/pass through any of the boundaries of the building removed a bit of the link to an actual real life building, making it feel like it the "game" didn't really have any purpose.” It seems that the ability to teleport through walls and doors is a double-edged sword. It does make it quicker to move around and experience the environment, at the cost of confusion if one is not familiar with the environment, and that it can lead to less immersion into the environment.

Judging based on observations; three things of note can be said. First, that while at first participants might be slow to move around as they get used to the controls, once they do get used to them, they often teleport rapidly through the virtual environment without much problem, even to make small adjustments. The reason teleportation was used for small adjustments is also because of the limited space testers had to move around
Secondly, quite a few users felt the need to move through the eHH as they would in real life, even insisting on opening and closing the doors. Even when they were told that this is not required, some testers still did it. This can be because of the immersion they felt in the environment. Getting used to VR can take a while, and at first many people would instinctively do things as they would in reality, even knowing that it is not needed.

Thirdly, it should be noted that not a single tester mentioned anything about cybersickness in their time in the environment. This is positive as it likely means that the teleportation did not result in cybersickness for the individuals. Comparatively to the first prototype, in which joystick movement was a possibility, it tended to make people sick as noted with Wezel (2019).

8.2.4.4 Open and Closing Doors:
Opening and closing doors had the lowest overall score amongst all the questions at a 4.29, with a standard deviation of 1.11, indicating that the difference in score can be quite significant.

There are likely multiple reasons for this. The first is that doors can only be opened in the direction they should be opened. This resulted in many people trying to push open a pull door, or vice versa. Further, it was challenging to determine rather a door was push or pull door. When doing this in the physical world, one would feel resistance trying to push open a pull door, and this is not the case in VR. There is no feedback to the user. On top of this, the doors above average size in the eHH. Participants were also limited in physical movement due to the small testing space. They had to try to open the doors whilst having little room to make the required movements with their arms and bodies. One user commented that “having to open doors through myself” was one of the weakest aspects of the environment.

8.2.5 Interaction with User Interface:
Interacting with the user interface scored a 6.14, one of the highest scores given. The standard deviation of 1.11 does indicate that the difference in this score can be quite significant. Further, not too much can be said about this, for most complains relevant to this fit better under interacting with the controller.

8.2.5.1 Cycling through different perspectives:
Cycling through the cameras is the only question in which ease of and the experience was asked. The reasoning for this is further explained in Appendix 3 – User Testing Results:. Regarding ease of use, it scored a 5.86, with a standard deviation of 1.46. Indicating that there was quite some deviation between people’s answers. This is further explored in the interacting with controller subsection. Comparatively,
users experience was scored at 5 with a standard deviation of 2. Showing that the results were even more varying. Such variation can likely be attributed to one user who had a rather severe fear of heights and gave their experience a score of 1. The cameras are placed 2.8m in the air, and suddenly seeing thing from this perspective can be very jarring with a severe fear of heights. It does highlight that it is beneficial to test the ease of use separately from experience.

Further, people were inclined to try and control their character while being in a CCTV camera perspective. This made it more difficult for them and seemed to confuse them; not realizing that this is not the intended use for it. Likely this can be attributed to this information not being specified. Participants can still move one's hands while experiencing things from a CCTV perspective but not interact with the environment.

8.2.5.2 Providing an Overview:

Users believed that the application was very insightful in providing the user with the layout of the eHH. This is indicated by the score of 6.43 and a low standard deviation of 0.53.

One tester mentioned that the tool gave them “the ability to look into and engage with the room as a more personal experience and sort of remember the layout in your own way (via a personalized experience so to speak)”. Likewise, another participant commented that “the 3D spacial [spatial] feeling is really close to the real one.”

It should be noted though, that observations did lead to noticing that testers seemed to struggle with grasping the eHH layout at first. This is also best summarized with a quote from a user; “I accidentally teleported into the bathroom while not knowing that there was one, and was confused if this was the next scene, since i accidentally pressed the teleport button instead of the select scene button. Again not necessarily a bad thing, but might be prone to user error if they are not familiar with the scene.” It would be beneficial to provide users with an overview of the eHH before they use the application. Seeing a map of the environment and having it explained what room is what. This tool can then build upon that experience by allowing users to experience the environment.

8.2.5.3 Interacting with Controller:

So far, analysis has been conducted in the order on which questions appeared in Table 3. This holds true, except for interacting with controllers. This element seemed to be the weakest link in user testing from observations, despite the fact that it scored a 5.29, with a standard deviation of 1.11. Testers’ comments and observations alter that view somewhat concur with the observations.
“It is possible to get the buttons confused, and then you suddenly change perspective without realizing and is it hard to get back.” An issue that seemed to be near consistent amongst all testers. As can be seen in Figure 7 on page 34, there are five buttons that can be pressed. Of those five buttons, the one that seemed to cause the least problems was the grab item button on the grip of the controller. This is likely due to it being the middle finger button. In order to press this button, a movement similar to closing one’s hand has to be made; making it a rather intuitive movement. It was also the most isolated button of all, not being close to any other button. The other button that seemed to be fine is the rotation using the joystick. It seemed to come rather naturally to people.

The real problem was between the interact with UI element, cycle through the cameras, and teleport button. The UI element and cycle through camera button feel identical and are placed closely next to one another. For an inexperienced user, to differentiate them, would require the user to look at the controller, which is not ideal. As users get more familiar with the controls, and they practice with the tool, this will likely stop being the case. Before it does, however, it can cause situations as the earlier mention tester who accidentally teleported through the wall instead of pressing the interact with UI button.

What seemed to cause the most confusion was between the interact with the UI element and the teleport function. When the ‘A’ or ‘X’ button is touched, a laser comes out of the controller to indicate where the user is aiming. This was made to let the user see where they would press, something which was noted to be a struggle (Wezel, 2019). Disabling this laser would result in a dot to appear on UI element when aiming instead of the large visible laser. However, when the user wants to teleport a similar, although different colour and an arched laser is also created. This can result in the user confusing the two buttons; teleporting when they want to press a UI element and pressing UI element (often with nothing happening) when they want to teleport.

8.2.5.4 Small testing environment:
The users were asked to test the tool in a small testing environment of around 1.3m by 1.3m. While this is not an ideal setting, it was the easiest to get everything functional and arguably most beneficial. Many people will be using such a tool in places where space is not plentiful, and thus it needs to be possible to use the said tool where space is sparse. This test revealed if caution is taken, it is possible to use a small environment. It will result in the user having to teleport more, even for small distanced, but everything still works.
8.3 Presentation to eHealth House Researchers:

A presentation to was given to some of the researchers, and staff whos work concerns the eHH. While they were not able to test the product itself at the time, they were informed about the tools functions and features. It was beneficial in gauging the researchers’ and staffs’ position and relation to the tool.

The researchers seemed very interested in the tool, mainly with interests regarding showcasing and preparation of research. A question that came up was by a researcher interested in knowing if the tool could be used to see if certain furniture would fit through the doorways. Researchers also showed interested in the ability to make their own custom environments and asked specifically about the ability to import custom items. Further, it has been asked if the tool can be showcased at an event at the Tech-Med Center in order to allow different researchers to try and use the tool.

When the tool was presented, a wide variety of staff and researchers used the tool. It was useful in providing an insight into how users would go about using the product. Since it was at a demo, participants could not try the tutorial, and thus struggled with getting the hang of the controls more than user testers. Testers would often confuse the cycle through CCTV camera button with the Interact with UI button. Further, when people wanted to press the interact with UI button, there was a tendency to press teleport instead, aligning with user tester strugglers.

In the demo, users had the ability to properly walk around in reality, having that movement translated in VR. Users could walk several meters in all directions. It should be noted, that for many, the use of VR is still very novel, resulting in their actions to be linked to reality, and not the VR environment. People would avoid ‘obstacles’, walk around furniture, or carefully step over fallen furniture, despite the space in front of them in reality being completely empty. When users realized they could walk through the wall, they would approach it very cautiously and very slowly put their hands and feet through it first. When attempting to pick up objects on the other side of a table or bed, they would stretch out or move their body in such a way to not touch the furniture bellow them, despite it not existing.

This shows that the environment is immersive for the users and treat it much the same as reality. This conforms with the result of user testing. Some people at the demo were surprised that they could teleport through walls and doors and were not sure why this was a possibility. Some others were inclined to navigate themselves through the doors regardless.

Before the demo was given, there seemed to be an issue with the scaling of furniture. Their sizes seemed not to be reflective of the size furniture had. Hence, it was quickly adjusted before the demo. This
seemed very strange as it should have been just fine. After the demo, in order to see why furniture seemed smaller then it should, several tests were conducted. Real-life distances were measured, such as the distance between the floor and the top of a desk, as well as the sizes of tables and sinks. These measurements were then compared to the measurements of similar pieces of furniture in VR. The way this was done was by placed the controllers at the edges of a piece of furniture, and measuring the distance between the controllers, as well as placing the controller on desks and seeing how close they were to the desks in the environment. For both measurements and comparison, their size in VR was very indicative of their size in real life. At worst there was a difference of no more than a couple of centimetres. Nevertheless, when in the virtual environment, it felt completely wrong, despite them being the right size.

Given this information, and the fact that people who have been in the eHH in person and the virtual app seem to nearly unanimously agree, and mentioned, that the virtual reality application seems larger than the actual eHH. It should be noted that Wezel (2019) mentioned this as well during the interview. The fact that people think this is strange, as it was already calculated (as mentioned in section 7.1.1) that the size is only off by about 0.82%, which should not be noticeable. This likely means that the way the headset renders the environment is such that it is not giving the user a proper perspective of the environment. This would only have been discovered had the demo taken place, by the feedback received from users, and by the observations made. Adjusting this can be difficult. The scale and field of view on the camera seem to be implemented in such a way where they cannot easily be changed. Changing their respective values, by manually altering the scale or field of view, will have the actions reset upon launching the program.

8.4 Evaluation Synopsis:

While the tool did not integrate SH3D with Unity like it initially set out to do, the resultant tool regardless manages to do quite a bit. Most of the features that were required are still included and can be used. Further, user testing has shown that most features seem relatively easy to use, and comment shows that the tool does help in providing an overview of the eHH in different settings.

The best way to see this tool is one that functions as a prototype to the desired tool. Instead of having a tool that can do everything that was initially intended, the provided tool gives an overview of what the resultant product might be like. In essence, ‘imagine a researcher created this scene in SH3D and imported it to this tool, this is what you will be able to do with it’. While this ‘prototype’ also includes the possibility to import custom scenes with a bit more difficulty, the furniture will not be movable.
It can also be said, that despite the tool not yet being ideal, it is something that the Tech-Med Center is looking forward to using and developing, potentially even making it an asset for the eHH. This would mean that improvements need to be made to the tool so that its shortcomings are resolved, or at least minimized. Namely, the rendering of the tool, the integration with SH3D, and the physics engine.
9 Improvements:

This section consists of three sub-sections. The first section explores improvements that need to happen regardless. The second and third are both ways in which the application could be improved to allow users to make their own custom environments properly. Smoother integration with SH3D has two subsections specifically for itself, as it is one of the main problems that need to be overcome in regards to improving the application.

9.1 Definite Improvements:

User testing and demonstrating the tool at the Tech-Med Center made it clear that several elements need changing or improvement. Two main things need to be improved in order to make the whole application come together more.

Firstly, the physics engine. Currently, when two objects are placed inside each other, they are rapidly propelled across the room. While it is understandable why Unity does this, it is not the most practical for the user. When trying to place a chair under a piano, misplacing it by one centimetre can result in the piano being launched across the room, leaving a mess of furniture in its wake. It makes the application seem unpolished and results in messing up the environment of the virtual eHH. Making a better physics engine that is more user-friendly is vital.

The second is what was highlighted during the demonstration. The way the user perceives the world is such that furniture and objects look smaller than they are, but the environment itself looks larger than it is. This can confuse the user and make the environment seem less accurate then it is. Fixing the way the environment gets rendered and presented to the user will result in the environment being more immersive and more reflective of reality, making it a necessary improvement.

Other improvements need to be made, as well. One of them is the double-edged sword of being able to teleport through walls and doors. Users provided insight that this feature provides both pros and cons. Being able to let the user choose; rather, they want to teleport through walls or not seems like the best option. This can be done by adding a toggleable UI element in the control room. If toggled on, the users will be able to teleport through the walls and doors, if toggled off, they will not be able to do so. This allows the user to alter the experience of this tool to their preferences.
Another toggle needs to be added to allow users to enable and disable the laser by touching the
interact with UI element button. The laser has caused confusion for users and allowing them to change it,
so it fits their individual needs, seems like the right course of action.

In a similar vein, providing a toggleable switch the enable and disable doors seems beneficial.
According to the users, they struggled most with the opening and closing of doors, spending quite some
time on it. Allowing them to enable or disable doors to their liking should make the program more user-
friendly. This will be done by placing a toggle in the control room.

Taking the recommendation from one user, making a distinction when the user is watching the
scene from a CCTV perspective. Adding “• REC” in the colour red in the corner of the user’s vision, along
with the name of the camera they are watching from should clarify that to users. This feature would be on
by default but can be disabled with a toggle, also found in the control room. This feature should be
toggable, as users can get annoyed by being forced to have a UI element imposed on them.

9.2 Improvements to Sweet Home 3D Integration:

One of the approaches it to stick with the original idea and improve the pipeline process between
SH3D. This would take some time to implement but should be possible, considering that SH3D is open
source. Altering the code so that it exports its objects differently should be possible and make the
integration with Unity far more functional. A possible way to do this is written in Appendix 4 – Possible
Solutions to Pipeline SH3D with Unity:

9.3 Improvements to Singular App

Another way to tackle this issue is to make the entire tool self-reliant. The user will never need to
change some aspects about the room. The shower, kitchen, toilet, sinks and the like are there to stay. This
means that it is possible to make the user do the entire process in Unity.

Split the app up in two parts. The first is done on a computer, not in VR. The user gets a top-down
view of the eHealth House and can select from furniture from a UI. This furniture is fully functional with the
environment and ready to be used. They can then place the furniture down where they want it. In essence,
recreating all the functionality of SH3D, and likewise, all the functionality of the Unity editor inside the
application.

This application, while looking ideal at first, does come with certain drawbacks. Simply doing the
above does not enable more advanced features, such as changing the material of objects, rotation,
elevation, nor scaling. Such features would need to be added as well. This is all already possible in SH3D. In order to integrate these features as well, a more advanced program would need to be written.
10 Conclusion:

This thesis looked at the entire process underwent for the development of a tool to aid researchers in planning and preparing their research at the eHealth House. The paper goes over the initial ideas produced and how they were going to be implemented. It goes on to analyze the problems that were faced and how this impacted the project. It provides information on what the final product has become, as well as a guide on how to use its more advanced features. The developed tool, while not ideal, does function as a prototype of what it would be like the work with the finished, ideal product.

The process to complete such a tool is more complex than initially thought. Along the way of development, multiple issues came up that function not only as a hurdle but also as a great learning experience how to handle situations when the initial plan does not work out. How to work when one has to adapt to one’s situation, the desires of a client, time limits, and one’s own limitations. In essence, what is taken away from this entire process is not merely the developed tool, but also a deeper understanding of issues that will be faced more often in the future. With this project came the learning patience due to having to painstakingly solve one small issue for hours on end. Also came learning to see the same idea from different angles which were taught by performing interviews and user reviews.

More importantly, the effort that is required in order to finish this product to its current functionality serves as a strong reminder of the effort and dedication that went into everything in one’s day to day life—functioning as a reminder of the simplicity of much in life that is often taken for granted. The reason that much of daily life is simple is due to people’s efforts and dedication to solve even the most mundane issues.

While things in life can be simple, and while the produced tool might be good, one should never think that things cannot become better. The tool provides the user with possibilities that were not there before — new ways to explore and experience the eHH. The provided tool can be used to help researchers in the planning and preparation of their research at the eHH. They can create their desired setup, and explore it in virtual reality. They can experience it from perspectives that would not be possible in real life. Despite this, the tool can be improved, and despite all it has to offer, it can be better.
11 Future Work:

This product is not ideal, and thus its further developments are always beneficial. Taking the current tool constructed as a reference, there are several paths it can go on. It can be developed further for helping to plan and prepare research, as is the original goal of the current tool. It can be developed more with showcasing in mind, a secondary goal of this tool. It can also be developed for experimentation.

The first is further development of the tool as is. This further development would aim to solve the shortcomings that this tool had, such as enabling a smooth SH3D to Unity integration. Then researchers can create their environment in SH3D and manipulate it in VR. It would be possible to develop a tool without the use of SH3D. Develop a tool in Unity in such a way that everything the researcher could want or need is in a standalone application, as discussed in 9.3. Regardless of which of the two versions it is, this tool would also need to integrate more of the feedback received from user testing and demo presenting, as discussed in section 9.1— such as solving the rendering and physics engine. It would also be possible to implement features so that the “should have”, “could have”, and “won’t have” from the requirements are included as well. This new tool should also need to conduct user testing, with a more extensive and more diverse group of people as was done in this paper. However, if fully functional, it will allow researchers to create unique settings of the eHealth House in a matter of minutes and explore those environments in 3D.

The second route a future tool can take is dedicated to adapting the tool for showcasing. As mentioned in the interview with Hermans (2019), the goal of such a tool would be to introduce people, and show them, what the eHealth House is, without them ever having been there. In such a case, two approaches can be taken. One in which the user can, in a matter of minutes, explore the eHH and get a basic understanding of the layout and what the eHH is. Another possibility is a tool in which the user experiences the eHH in more depth, where the environment has more details and more interactivity. For a conference, the simpler one would most likely be more beneficial, allowing many people to get an overview quickly. For introducing the eHH to partners, investors, stakeholders, and clients, the latter might be of more benefit, really allowing the user to understand the benefits of this environment. Both would want to have an improvement in overall functionality, such as graphics, physics engine, and rendering.

The third and most difficult future work of this tool stems from the interview with Wezel (2019). In this case, the patient would be put into the VR environment for testing. This can be done for both preliminary testing, or to test how VR compares to reality. Such a tool would be the most challenging (and
with current technology, not feasible) implementation of the tool. Realism would require a massive improvement and certain functionalities need to be changed. The user should not be able to teleport, but rather, must walk around the entire environment; meaning the user should be able to walk the full size of the eHH. It would likely have to function with a combination of both VR and AR (augmented reality). Lastly, all the data must be recorded for analysis and research. Data such as, when the user is where, what they are looking at, how they are moving, and the relation of the user in respect to the rest of the room. It would provide researchers with an extensive amount of data to work with.

The Techmed Simulation Center (TSC), under which the eHealth House falls, has shown lots of interest in integrating and using virtual reality for a wide variety of purposes. Likewise, the use of VR technology can be very beneficial to them. However, the created tool is in its infancy and requires more development before it properly becomes viable. With investment and development, an improved version of this tool can be made to further the goals of TSC. While several recommendations are made for possible improvements to the tool, and directions it can go, there are plenty of other options available to be explored, some of which might not be apparent now. As my favourite character, Hinata Miyake said: “Never think the stars you see are all the stars there are” (Ishizuka, 2018).
References:


Hermans, M. (2019, November 27) Interview with the Content Coordinator of the eHealth House/Interviewer: G. Hexspoor.


Appendix 1 – List of available software:

<table>
<thead>
<tr>
<th>Name of Program</th>
<th>Type of Program</th>
</tr>
</thead>
<tbody>
<tr>
<td>ArchiCAD</td>
<td>Commercial</td>
</tr>
<tr>
<td>Assetforge</td>
<td>Commercial</td>
</tr>
<tr>
<td>AutoCAD Civil 3D</td>
<td>Commercial</td>
</tr>
<tr>
<td>Blophome</td>
<td>Freemium</td>
</tr>
<tr>
<td>CorelCAD</td>
<td>Commercial</td>
</tr>
<tr>
<td>DreamPlan</td>
<td>Freemium</td>
</tr>
<tr>
<td>Floorplanner</td>
<td>Freemium</td>
</tr>
<tr>
<td>Home Designer</td>
<td>Commercial</td>
</tr>
<tr>
<td>HomeByMe</td>
<td>Freemium</td>
</tr>
<tr>
<td>Live Interior 3D Pro</td>
<td>Commercial</td>
</tr>
<tr>
<td>Magicplan</td>
<td>Freemium</td>
</tr>
<tr>
<td>pCon.planner</td>
<td>Free</td>
</tr>
<tr>
<td>Planner 5D</td>
<td>Freemium</td>
</tr>
<tr>
<td>Roomeon</td>
<td>Free Personal</td>
</tr>
<tr>
<td>Roomstyler</td>
<td>Free</td>
</tr>
<tr>
<td>SketchUp</td>
<td>Freemium</td>
</tr>
<tr>
<td>Sweet Home 3D</td>
<td>Freemium (Open Source)</td>
</tr>
</tbody>
</table>

In order to have a consistent form of judging what type of program these are; this is all according to https://alternativeto.net/ Johansson and Olausson (2019).
Appendix 2 – Advanced User Guide – How to Import Custom Furniture:

Getting Started:

In order to import your own custom environments, the following is needed:

Sweet Home 3D: [http://www.sweethome3d.com/](http://www.sweethome3d.com/)

Unity: [https://unity.com/](https://unity.com/)

Creating Custom Environment:

Open the application Sweet Home 3D.

In the top left corner, press file, open, and locate the file “eHealtHouseEmpty.sh3d” and open it.

Go to File->Save As, and save it with a different name, such as “Custom Environment 1”.

The Workspace:

The screen will be divided into four sections. Top left has a list off all the furniture that can be used.

The top right is a top-down view of the eHealth House. This is where furniture will be placed.

Bottom right is a 3D view of the current setup.

Bottom left is a list that contains all the furniture in the eHealth House.
Different Types of Furniture Lists:

There are two types of way to display the furniture, “Category Folder” and “Searchable List”, as can be seen in Figure 23 and Figure 24.
Changing way of seeing furniture. File->Preferences (Figure 25)

And in this new window there are two options to select (Figure 26)
Figure 25: File -> Preferences

Figure 26: Preferences List
Adding Furniture:

By clicking any piece of furniture in the furniture list (top left) and dragging it to any location on the top down view (top right) will create that piece of furniture. It is possible to move this anywhere you want in the eHealth House. The newly created item will now also be visible on the furniture list on the bottom left.

Adjusting Furniture:

It is also possible to change elements of the furniture selected. By double clicking a piece of furniture, a new window called “Modify Furniture” appears. This can be used to change elements of the furniture.

![Modify Furniture Window](image)

*Figure 27: The modify furniture window*

There is quite a bit of elements that can be added.

**Name:**

Name: One can change the name of the object.
Location:
X and Y: The location indicates where the item is located.
Mirror shape: allows the item to be mirrored along an axis of symmetry
Part of base plan: makes item part of base plan (can be useful once Sweet Home 3D and Unity are integrated better)

Orientation:
Angle: allows the user to rotate the object on the z axis (the one to be used most often)
X axis: allows the user to rotate the object on the x axis
Y axis: allows the user to rotate the object on the y axis

Size:
Width: Change the width of the object
Depth: Change the depth of the object
Height: Change the height of the object
Keep proportions: ensures that the scale of the width to depth to height remains constant. Changing one of those values will have a proportional effect on the other values.
Modify Posture: *Explained later-on*

Shininess:
Unchanged: Default shininess
Matt: Less shiny
Shiny: More Shiny

Colour and texture:
Colour: change everything in the object to the selected color
Texture: change everything in object to the selected texture
Materials: *Not always available - Explained later*

Visible:
Visible: is the object visible or not
Modify Posture:

This option is only available if the posture / object can be modified.

Modify posture can be used in order to alter states of an object. For example, opening and closing doors, or changing the pose of a mannequin as illustrated in Error! Reference source not found.. The way this can be altered is by clicking and dragging on the piece.

![Figure 28: Different possible poses using the modify posture function](image)

Materials:

When opting to change the materials of an item, by pressing modify the furniture materials window pops up, as seen in Figure 29. Here it is possible to select distinct parts of the item and change and edit the colour, visible-ness, texture and shininess for each component individually.
Exporting the Application:

In order to export one’s scene, one must first select all the furniture that needs to be exported. Click the topmost furniture piece, and shift click the bottom most furniture piece. IMPORTANT: Make sure that the “_DO NOT EXPORT” item is not selected; as seen in Figure 30.
In the top left menu, press “3D view” -> “Export to OBJ format”.

Create a new folder and save the file inside the new folder under a name of choice. When save is pressed, a new window will pop up, chose “Export Selection”.

**Importing it to Unity3D:**

<table>
<thead>
<tr>
<th>Name</th>
<th>Width</th>
<th>Depth</th>
<th>Height</th>
<th>Visible</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wood lower ca...</td>
<td>1,695</td>
<td>510</td>
<td>860</td>
<td>✔️</td>
</tr>
<tr>
<td>Wood glass do...</td>
<td>452</td>
<td>392</td>
<td>300</td>
<td>✔️</td>
</tr>
<tr>
<td>Wood glass do...</td>
<td>452</td>
<td>392</td>
<td>300</td>
<td>✔️</td>
</tr>
<tr>
<td>TV shelf unit</td>
<td>1,300</td>
<td>444</td>
<td>333</td>
<td>✔️</td>
</tr>
<tr>
<td>TV shelf unit</td>
<td>1,300</td>
<td>444</td>
<td>333</td>
<td>✔️</td>
</tr>
<tr>
<td>TV</td>
<td>989</td>
<td>181</td>
<td>648</td>
<td>✔️</td>
</tr>
<tr>
<td>Table</td>
<td>1,511</td>
<td>804</td>
<td>775</td>
<td>✔️</td>
</tr>
<tr>
<td>Sofa</td>
<td>2,000</td>
<td>865</td>
<td>880</td>
<td>✔️</td>
</tr>
<tr>
<td>Plant</td>
<td>818</td>
<td>690</td>
<td>1,366</td>
<td>✔️</td>
</tr>
<tr>
<td>Pillow</td>
<td>399</td>
<td>194</td>
<td>366</td>
<td>✔️</td>
</tr>
<tr>
<td>Pillow</td>
<td>399</td>
<td>194</td>
<td>366</td>
<td>✔️</td>
</tr>
<tr>
<td>Pillow</td>
<td>344</td>
<td>167</td>
<td>316</td>
<td>✔️</td>
</tr>
<tr>
<td>Piano stool</td>
<td>356</td>
<td>376</td>
<td>350</td>
<td>✔️</td>
</tr>
<tr>
<td>Piano stool</td>
<td>438</td>
<td>462</td>
<td>400</td>
<td>✔️</td>
</tr>
<tr>
<td>Piano stool</td>
<td>542</td>
<td>572</td>
<td>450</td>
<td>✔️</td>
</tr>
<tr>
<td>Induction cook...</td>
<td>560</td>
<td>495</td>
<td>4</td>
<td>✔️</td>
</tr>
<tr>
<td>Gray cabinet w...</td>
<td>800</td>
<td>620</td>
<td>990</td>
<td>✔️</td>
</tr>
<tr>
<td>Gray cabinet</td>
<td>800</td>
<td>620</td>
<td>860</td>
<td>✔️</td>
</tr>
<tr>
<td>Gray cabinet</td>
<td>800</td>
<td>620</td>
<td>860</td>
<td>✔️</td>
</tr>
<tr>
<td>Gray cabinet</td>
<td>800</td>
<td>620</td>
<td>860</td>
<td>✔️</td>
</tr>
<tr>
<td>Fridge</td>
<td>600</td>
<td>681</td>
<td>1,800</td>
<td>✔️</td>
</tr>
<tr>
<td>Floor lamp</td>
<td>313</td>
<td>909</td>
<td>1,800</td>
<td>✔️</td>
</tr>
<tr>
<td>Chair</td>
<td>444</td>
<td>584</td>
<td>820</td>
<td>✔️</td>
</tr>
<tr>
<td>Chair</td>
<td>444</td>
<td>584</td>
<td>820</td>
<td>✔️</td>
</tr>
<tr>
<td>Chair</td>
<td>444</td>
<td>584</td>
<td>820</td>
<td>✔️</td>
</tr>
<tr>
<td>Chair</td>
<td>444</td>
<td>584</td>
<td>820</td>
<td>✔️</td>
</tr>
<tr>
<td>Carpet</td>
<td>2,000</td>
<td>2,250</td>
<td>10</td>
<td>✔️</td>
</tr>
<tr>
<td>Bowl</td>
<td>401</td>
<td>401</td>
<td>204</td>
<td>✔️</td>
</tr>
<tr>
<td>Books</td>
<td>302</td>
<td>213</td>
<td>339</td>
<td>✔️</td>
</tr>
<tr>
<td>Armchair</td>
<td>696</td>
<td>694</td>
<td>949</td>
<td>✔️</td>
</tr>
<tr>
<td><strong>DO NOT EXP...</strong></td>
<td><strong>9,372</strong></td>
<td><strong>12,408</strong></td>
<td><strong>2,400</strong></td>
<td>✔️</td>
</tr>
</tbody>
</table>

*Figure 30: Example of selected items.*
In order to import it into Unity, Unity must first be opened. Open Unity Hub, and press “ADD”. Navigate to the download location of the Unity App and select the folder: “UNITY_VReHealthHouseTool”. Open this application.

**Workspace:**

![Unity Workspace](image)

Figure 31: Unity Workspace

To import the environment by dragging the earlier created folder (during the export process) into the Assets bar on the bottom of Unity.

After importing the furniture, in the same Asset bar navigate to Scenes->Final Scenes->“EmptyFurniture2”, and open it by double clicking it.

**Importing it to the scene:**

Navigate back to the assets main folder (this can be done by pressing “Assets” in the top left of the Assets bar – blue rectangle in Figure 31).

Open the imported folder and find the prefab that is created. It is likely the second item and it will most likely have an arrow on the right side of the object. In the case of Figure 32, the green square.

![Imported Folder](image)

Figure 32: Example of what the inside of the imported folder can look like

Take that item and drag it onto the eHealth House in onto the scene (pink rectangle), or into the hierarchy (yellow rectangle).
Fixing size and Location:

The only problem now is that the environment is too large, and not in the right location. In the hierarchy (yellow rectangle), select the newly imported object (should be the only object with a blue cube next to its name).

In the inspector, on the right-hand side (cyan rectangle), change the location to 0, 0, 0, and the scale to 0.01, 0.01, 0.01 as seen in Figure 33.

Figure 33: Inspector settings

Done! Press play in the top center (above the pink rectangle), and you should be able to explore your environment in VR!
Appendix 3 – User Testing Results:

This appendix will delve into the user testing process. It should be mentioned, that in the process several questions were rather unclear, allowing them to be interpreted as two different questions. Some people answered it as question A, and others as Question B. Examples of this:

“How easy was it to press buttons?” Can be interpreted as ease of pressing buttons on the controller, or the intended UI element buttons.

The second question that had this problem was: “How was your experience with rotating through different camera angles?” There were two problems with this. The first being that rotating through camera angles could be interpreted as using the thumb stick to rotate the player, but it could also be interpreted, which was intended, to cycle through ceiling cameras. Further, this was the only question pertaining to experience and not ease of use, which was originally going to be test.

One user answered ‘1/7’ on experience though. In order to not infringe the academic integrity of this paper, and not to commit acts that can be seen as academically fraudulent, the following was done. Users were asked to fill in a secondary form, in which they would answer all those questions: Ease of use of pressing buttons on the controller (thus also how easy it was to rotate), ease of use of pressing UI element buttons, ease of use of pressing to cycle the user through CCTV camera perspectives and their experience in doing so.

The aim was to provide an honest and transparent as possible solution to the problem. Tester’s names have been hidden, and their age removed for privacy reasons.

Before the users could answer any questions, they all consented to the following. It was required to do the user testing.

“You are about to undertake a user-testing of a Virtual Reality (VR) tool developed by Giomar Hexspoors for his Graduation Project of Creative Technology. I understand that while some personal data is collected in this user testing, all of it will be anonymized for the report. I also agree that I can withdraw my participation at any time without having to give a reason for said withdrawal. Upon doing so, the user testing will end immediately and all relevant data will be deleted.
The VR tool is developed in order to help researchers plan and prepare their research at the eHealth House; a simulated living environment allowing researchers to observe their patients. In order to check the feasibility of this tool, its features, and ease of use, several tests have been created which you are about to undergo. Prior to this, some preliminary data will be collected.

If you have any questions, feel free to ask at any time.”
Individual Responses:

Table 4: Table of responses for user testing – Likert Scale questions

<table>
<thead>
<tr>
<th>Question</th>
<th>Subject 1</th>
<th>Subject 2</th>
<th>Subject 3</th>
<th>Subject 4</th>
<th>Subject 5</th>
<th>Subject 6</th>
<th>Subject 7</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>Stage 2:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>The participants were asked to do the tutorial and were then asked the following questions</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I believe the tutorial was intuitive?</td>
<td>3</td>
<td>6</td>
<td>5</td>
<td>7</td>
<td>4</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>I believe the tutorial was beneficial in helping to understand what I can do?</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>7</td>
<td>5</td>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td>Stage 3:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>After the tutorial was completed, and the previous questions were answered, they had to complete the tasks as mentioned earlier. Once completed, were asked to fill this out.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>How easy was it to pick up and move furniture?</td>
<td>7</td>
<td>6</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td>How easy was it to get around the eHealth House?</td>
<td>5</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>5</td>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td>How easy was it to open and close doors?</td>
<td>6</td>
<td>4</td>
<td>3</td>
<td>5</td>
<td>4</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>How easy was it to press buttons?*</td>
<td>5</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>4</td>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td>How was your experience with rotating through different camera angles?*</td>
<td>4</td>
<td>7</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>7</td>
<td>1</td>
</tr>
<tr>
<td>If you were to go to the eHealth House, how well of an overview of the environment do you think you’ll have?</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>7</td>
<td>6</td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>Question</td>
<td>Subject 1</td>
<td>Subject 2</td>
<td>Subject 3</td>
<td>Subject 4</td>
<td>Subject 5</td>
<td>Subject 6</td>
<td>Subject 7</td>
</tr>
<tr>
<td>-------------------------------------------------------------------------</td>
<td>-----------</td>
<td>-----------</td>
<td>-----------</td>
<td>-----------</td>
<td>-----------</td>
<td>-----------</td>
<td>-----------</td>
</tr>
<tr>
<td>How easy was it to interact with the User Interface?**</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td>How easy was it to interact with the controller?**</td>
<td>4</td>
<td>4</td>
<td>6</td>
<td>7</td>
<td>5</td>
<td>6</td>
<td>5</td>
</tr>
<tr>
<td>How easy was it to cycle through different perspectives?**</td>
<td>5</td>
<td>3</td>
<td>6</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>6</td>
</tr>
<tr>
<td>How was you experience cycling through different perspectives?**</td>
<td>6</td>
<td>5</td>
<td>6</td>
<td>6</td>
<td>4</td>
<td>7</td>
<td>1</td>
</tr>
</tbody>
</table>

* Indicates that the question has been null and voided due to unclarity issues

** Indicate that the question replaced unclear questions
Table 5: Table of responses for user testing – Written questions and multiple choice questions

<table>
<thead>
<tr>
<th>Question</th>
<th>Subject 1</th>
<th>Subject 2</th>
<th>Subject 3</th>
<th>Subject 4</th>
<th>Subject 5</th>
<th>Subject 6</th>
<th>Subject 7</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Have you ever used VR before</strong></td>
<td>I have limited experience with VR, (e.g, having used it once or a couple of times )</td>
<td>I have limited experience with VR before</td>
<td>I have very familiar with VR (e.g, heavy use of VR, or have developed VR applications )</td>
<td>I have limited experience with VR, (e.g, having used it once or a couple of times )</td>
<td>I have limited experience with VR, (e.g, having used it once or a couple of times )</td>
<td>I have moderately familiar with VR (e.g, using it every now and then)</td>
<td></td>
</tr>
<tr>
<td><strong>Have you ever seen or been in the eHealth House</strong></td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td><strong>Stage 2:</strong> Participants were asked after completing tutorial, before starting tasks</td>
<td><strong>Any things that you found very good about the tutorial? (optional)</strong></td>
<td>linear path of the tutorial makes it impossible to get lost spacially-wise</td>
<td>Slow buildup, easy steps to slowly get familiar with the system</td>
<td>Makes the user aware of what they can and cant do in the environment.</td>
<td>It explores different functions of the controllers.</td>
<td>The VR environment was convincing, interacting with the objects within it was intuitive. The door physics and such worked exactly as expected without</td>
<td>if you followed it neatly, it could be done really quickly, though still guaranteeing that the user experienced the things that they can do.</td>
</tr>
<tr>
<td>Question</td>
<td>Subject 1</td>
<td>Subject 2</td>
<td>Subject 3</td>
<td>Subject 4</td>
<td>Subject 5</td>
<td>Subject 6</td>
<td>Subject 7</td>
</tr>
<tr>
<td>----------</td>
<td>-----------</td>
<td>-----------</td>
<td>-----------</td>
<td>-----------</td>
<td>-----------</td>
<td>-----------</td>
<td>-----------</td>
</tr>
<tr>
<td>Anythings you found bad / deserved improvement about the tutorial? (optional)</td>
<td>Having to be told.</td>
<td>Maybe the orientation of some signs. E.g. I do not look up a lot and because of that found myself lost for a moment.</td>
<td>Perhaps have the sign above the door pop up after the color sorting</td>
<td>Spell check (bellow)</td>
<td>I had to read the text a few times to understand what buttons had what functions. And I accidentally teleported through the door.</td>
<td>Nothing in particular.</td>
<td>Could be a little bit more entertaining, in the sense that there isn't much exploring the feeling of the game since it is very on tracks (basically more stuff like the cube solving gimmick)</td>
</tr>
</tbody>
</table>

Stage 3:
Users were asked after finishing tasks
<table>
<thead>
<tr>
<th>Question</th>
<th>Subject 1</th>
<th>Subject 2</th>
<th>Subject 3</th>
<th>Subject 4</th>
<th>Subject 5</th>
<th>Subject 6</th>
<th>Subject 7</th>
</tr>
</thead>
<tbody>
<tr>
<td>What do you believe were the strongest aspects of the VR experience?</td>
<td>The user gets a proper feel of the environment they are being put into</td>
<td>The 3D spacial feeling is really close to the real one (except the weight aspect etc.)</td>
<td>Loved the physics</td>
<td>- The ability to look into and engage with the room as a more personal experience and sort of remember the layout in your own way (via a personalized experience so to speak) T-he ability to more intuitively place and reconstruct the environment to your liking.</td>
<td>After you get the hang of it, there are a lot of possibilities of moving around and designing a place. It could be applied to any space.</td>
<td>Interacting with the objects within the scene was quite fun, especially since the objectives were a bit silly.</td>
<td>It is more comfortable to phase around the place like a magical entity than it would ever be to just see it in real life. Plus, it adds a sense of comfort in not HAVING to do any effort to do the tasks that in real life would be a difficult and heavy chore.</td>
</tr>
<tr>
<td>Question</td>
<td>Subject 1</td>
<td>Subject 2</td>
<td>Subject 3</td>
<td>Subject 4</td>
<td>Subject 5</td>
<td>Subject 6</td>
<td>Subject 7</td>
</tr>
<tr>
<td>----------</td>
<td>-----------</td>
<td>-----------</td>
<td>-----------</td>
<td>-----------</td>
<td>-----------</td>
<td>-----------</td>
<td>-----------</td>
</tr>
<tr>
<td>What do you believe were the weakest aspects of the VR experience?</td>
<td>Nothing very trivial...but it does take some time to get a feel for which buttons complete which function</td>
<td>the fact that you can only move &quot;binary&quot; no movement except point-and-click allowed</td>
<td>Also the physics and having to open doors through myself</td>
<td>The weird interactions that objects have with each other, as well as the fact that you are able to teleport through walls. That is not necessarily a bad thing, but (in this case) makes it a bit more confusing to where you go and where you came from. e.g. I accidentally teleported into the bathroom while not knowing that there was one, and was confused if this was the next scene, since I accidentally pressed the teleport button instead of the select scene button. Again not necessarily a bad thing, but might be prone to user error if they are not familiar with the scene. It does</td>
<td>It is possible to get the buttons confused, and then you suddenly change perspective without realizing and is it hard to get back. And the glasses sometimes slid down a bit because they are heavy.</td>
<td>Physics still were a bit glitchy, passing out if you place objects within each other.</td>
<td>The ability to look/pass through any of the boundaries of the building removed a bit of the link to an actual real life building, making it feel like it the &quot;game&quot; didn't really have any purpose.</td>
</tr>
<tr>
<td>Question</td>
<td>Subject 1</td>
<td>Subject 2</td>
<td>Subject 3</td>
<td>Subject 4</td>
<td>Subject 5</td>
<td>Subject 6</td>
<td>Subject 7</td>
</tr>
<tr>
<td>-------------------------------------------------------------------------</td>
<td>---------------------------------------------------------------------------</td>
<td>---------------------------------------------------------------------------</td>
<td>---------------------------------------------------------------------------</td>
<td>---------------------------------------------------------------------------</td>
<td>---------------------------------------------------------------------------</td>
<td>---------------------------------------------------------------------------</td>
<td>---------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| What are your recommendations for improvement?                           | not allowing the objects go through each other or at least not make them fly everywhere once they do (unless it is "a funny mode" for you to play :) | Spell checking, physics interaction and maybe teleporting through walls, although again that could be debatable whether thats' a good or bad thing, since its prone to user error. | maybe make clearer difference between the different perspectives, maybe through colorcoding or so. | Removing the ability to teleport through walls, encouraging the player to have move around the place a bit more.
Appendix 4 – Possible Solutions to Pipeline SH3D with Unity:

A possible way to allow pieces to be automatically grouped once imported into Unity is by changing the name of the exported item. This is done by adding prefixed to the name. Exporting it with such prefixes will result in the grouping of items to be doable by script. One way of doing this is by giving each exported object two prefixes; “X_Y_Name”. Where X is either a 0 or 1 based on whether the exported object is “part of base plan” or not. This way, each object will start with an indicator if this object needs to be movable or not.

Let Y be reference to what piece of furniture it belongs to (i.e. the first piece of furniture will have Y of ‘0’, while the second piece will be ‘1’ – starting with 0 as an array does while coding). This way the items do not come pre-grouped, however, groups can be derived from their names relatively easy.

A script can first group all the object depending on if they are movable or not (the X variable), and then further group objects based on what piece of furniture they belong to (the Y variable). Relevant scripts can then be added to the parents of furniture objects.

There is, however, another way possible. Maya allows for objects to be exported in OBJ format, but also contains a featured called “Send to Unity”. When using “Send to Unity”, the resultant file, also in OBJ format, when opened in Unity comes pre-grouped. This indicates that it is possible to export the objects in such a way, that not much user involvement is required, and there is no need for very advanced scripts like in the previous solution. If that is the case, a script would only need to add the required components to the pieces of furniture, with no need to group them. This report cannot comment on the difficulty to implement this feature to Sweet Home 3D, it should, in theory, be a possibility. It is, however, possible that both these solutions require lots of changes to be made on the side of Sweet Home 3D, so this report cannot attest for how practical this solution is.