Computer Science

COC251

B621139

Health and Safety Virtual Reality Training Application

by

Kareem Ahmed

Supervisor: Dr F. Batmaz

Department of Computer Science
Loughborough University

April 2019

Table of contents

1 Table of Figures 5
2 Introduction 6
2.1 Brief 6
2.2 Aims and Objectives 6
3 Requirements 7
3.1 Literature review 7
3.1.1 VR Training 7
3.1.2 Health and Safety in VR 10
3.1.3 The online training VR environment 10
3.1.4 Movement in VR 11
3.1.5 Map editors 13
3.1.6 What The Health and Safety Department want? 17
3.2 Optional Background Research 18
3.2.1 Limitations in VR 18
3.2.2 Motion sickness in VR 23
3.2.3 VR Immersion 26
3.2.4 VR Design Principles 29
3.3 Existing tools 35
3.3.1 Authoring Tool 35
3.3.2 Virtual Reality Application 35
3.4 Existing technology 35
3.4.1 Mobile 36
3.5 Requirements table 36
4 Design 39
4.1 System Design 39
4.1.1 Authoring tool and Mobile application combined 39
4.1.2 Authoring tool and Mobile application separate 40
4.2 Authoring tool design 41
4.2.1 Canvas 41
4.2.2 Adding Objects 44
4.2.3 Adding Tasks 44
4.2.4 Adding Sounds 46
4.2.5 UI design 47
4.2.6 Authoring Tool design decisions 49
4.3 Database Design 50
4.4 Application Design 52

4.4.1 Menu 52
4.4.2 VR environment 52
4.4.3 Movement within the application 52
5 Implementation 53
5.1 Scope 53
5.1.1 Features that must be added 53
5.1.2 Features that were added that were not required 56
5.1.3 Features that were not added..57
5.2 Communication Overview 58
5.2.1 Authoring tool pushing data to the database 58
5.2.2 Authoring tool requesting data from the database...............................58
5.2.3 VR Application requesting data from the database 60
5.3 Data structures 60
5.4 Code Breakdown and Highlighting 62
5.4.1 Authoring Tool 62
6 Evaluation...66
6.1 Requirements 66
6.2 Test plan 66
6.2.1 Requirements Testing..66
6.3 Test results 66
6.3.1 Requirements Test Results 67
7 Conclusion 76
7.1 Success Criteria 76
7.2 Evaluating Success 76
7.2.1 Background research...76
7.2.2 Designing of the framework..77
7.2.3 Design of the deliverables 77
7.2.4 Implementation of the designs 77
7.2.5 Evaluation of the deliverable 78
7.2.6 Overall Success...78
7.3 Future Work...78
7.3.1 Move to 3D 78
7.3.2 Unlimited amount of rooms...78
7.3.3 Upload the users' objects or sounds 79
7.3.4 Better looking menu..79
7.3.5 The objective screen upon entering the room 79
7.3.6 Account system 79
7.3.7 Flow chart tasks...80

8 Appendixes 81
8.1 Health and Safety Storyboard 81
9 References 98

1 TABLE OF FIGURES

Figure 1MIST-VR simulator (Kanumuri, et al., 2008) 7
Figure 2 Example of comparable SP and VR victims. SP = standardized patient; VR = virtual reality (Andreatta, et al., 2010) 8
Figure 3 General view inside VR stope (Squelch, 2001) 8
Figure 4 Picture of Creator-AVR (EON Reality, 2019) 11
Figure 5 DOTA mod for Warcraft 3 13
Figure 6 G.E.C.K Creation Environment 14
Figure 7 Halo Forge Creation Environment 15
Figure 8 Super Mario Maker Creation Environment 16
Figure 9 Line Rider Creation Enviroment 16
Figure 10 Beat Saber Menu 17
Figure 11 Virtual Boy Headset 19
Figure 12 HTC Vive Disposable Hygiene Cover 20
Figure 13 Uncanny Valley 29
Figure 14 Accounting 30
Figure 15 VR Viewing Directions 31
Figure 16 VR Chat 32
Figure 17 Skyrim VR 33
Figure 18 Thalamic Labs Myo 34
Figure 19 System diagram for one application 40
Figure 20 System Diagram for separate applications 41
Figure 21 Chessboard Centreboard design 42
Figure 22 Empty canvas design 43
Figure 23 3D Canvas design 44
Figure 24 Task List Design 45
Figure 25 Flow Chart Design 46
Figure 26 First UI Design 47
Figure 27 Second UI Design 48
Figure 28 Popup Menu 48
Figure 29 Third UI Design 49
Figure 30 Adding Objects Example 54
Figure 31 VR Teleporter example 56
Figure 32 Array Overview 60
Figure 33 Objects Task Creation 63
Figure 34 Chessboard Task Creation 64
Figure 35 Nothing Task Creation 64
Figure 36 Sound Task Creation 65

2 INTRODUCTION

2.1 BRIEF

The project creates an online tool which allows academics to make VR environments with no prior knowledge in game development or coding. Loughborough's Health and Safety department has asked for this project to be created; they will be one of the stakeholders for this project. The project is broken down into three parts. The first part is the Authoring Tool; this tool allows the user to design an environment to their liking flexibly. The next part is a database which holds the designs that the users create. The final part is a VR application that takes what the user made on the authoring tool and translates it to the VR environment. The way all these works together and the parts themselves will be the deliverable for the project. To confirm that the deliverable to an appropriate standard for the Health and Safety department they have given a storyboard to follow as to what environment they want to be made with the tool. Successful replication of this storyboard in the VR environment by using the Authoring Tool will be considered a success.

2.2 AIMS AND OBJECTIVES

The project aims to create a tool that allows designing and playing of VR environments for learning purposes made by academics.

The objectives achieve this goal are.

1. Literature Review into the area
2. Designing the framework of the deliverable
3. Design the online authoring tool, database and mobile application
4. Implement the designs of the three deliverables.
5. Test and evaluate all the deliverables

3 REQUIREMENTS

In this section are the requirements for the project. This section starts with the literature review which will go through all the initial reading and research before any designing or implementation will happen. After the literature review, the report will go through the existing tools and technology that are already on the market. They are used as starting points and inspiration for the project.

3.1 LITERATURE REVIEW

3.1.1 VR Training

VR is becoming a training tool in many areas such as medicine where it has been "a potential, viable solution for junior endoscopists, for overcoming the "fulcrum effect", in a replicable, safe learning environment which allows objective and reliable quantification of skill levels by trainers." (Gallagher, et al., 1999). The use of VR for endoscopists can be seen in the MIST-VR simulator depicted below.

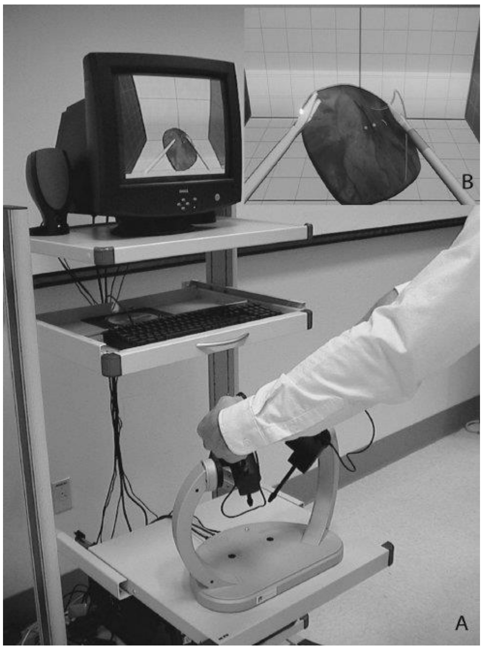


Figure 1MIST-VR simulator (Kanumuri, et al., 2008)

Another area VR is used is in disaster preparedness where it "provides flexible, consistent, on-demand training options, using a stable, repeatable platform essential for the development of assessment protocols and performance standards." (Andreatta, et al., 2010)

Figure 2 Example of comparable SP and VR victims. SP = standardized patient; VR = virtual reality (Andreatta, et al., 2010)

Moreover, in the mining industry VR "has the potential to provide effective training systems that are relevant to the South African mining industry" (Squelch, 2001) below is an image from the mining simulator used in the testing.

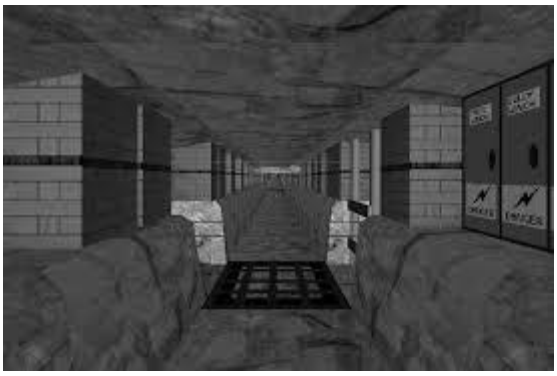


Figure 3 General view inside VR stope (Squelch, 2001)

This research shows that VR has uses in any industry or environment. The crossover between the South African mining industry and endoscopists is minimal, yet both can be accurately and successfully replicated using VR. Secondly VR provides a "safe learning environment", in all these examples the industry is either dangerous to oneself or others, especially during the training process. VR training is safe, as it can carried out in any environment and negates any dangers that may otherwise occur in training.

Next is VRs ability to be "replicable" and "repeatable" these training environments always start and end the same, they follow similar paths and put forth clearly to the user the knowledge that must be acquired. Since they are computer programs this happens consistently, there can be placed "assessment protocols and performance standards" on the users of the software to see if they have passed the training.

Lastly, these forms of training can be "on-demand", since the software is run on VR devices the training can be run at home. Most of the software is self-explanatory and does not require a master to run through the training this makes it the perfect option for people to do training at home on their own time instead of hiring out expensive venues or expensive setup to do the training.

(Hsu, et al., 2013) argues that VR "can be tailored to specific users as well as organizations" this means if the customer wants a more action-packed and exciting training program or lots of user interaction that's possible with the same software the only limiting factor is the customer's imagination.

the VR software "allow trainees to work at their own pace" with each individual in control of their learning environment, this is not possible in traditional training However it still, "is possible to apply Virtual environments to groups. Using a control panel actions can be set into motion providing a setting for group discussion and analysis" an interesting interaction which changes the environment on the fly allowing for a different experience each time.

VR is "a practical alternative that incorporates realism at a fraction of the cost of real-life exercises when considering the number of potential learners, range of applications and repeat scenario use." The reduction in cost is one of the reasons, so many different industries are looking into VR training as the reduction in fiscal budgets are reducing, and cuts need to be made. Although VR training may be expensive as an upfront cost, it is a onetime cost that once paid only needs little maintenance over time.

Lastly "VR-based platforms support data and video capture of time and critical action elements" this allows users to accurately and quickly record what users do in these training environments, which first can allow the creators of the environment to improve it by looking at the issues that are faced inside the software. The second allows the customer to look at and analyse their performances inside the simulation and apply that to the real-life scenario.

It is not all positives when it comes to VR training. The same journal also gave some disadvantages to the idea. The first is the "lack of familiarity with VR applications among disaster planning leadership may be a significant barrier to adopt such technology". What this means is the people who make the decisions on what training should be done usually are not well versed In the world of virtual reality let alone training simulations in virtual reality, so the chances of software like this put into practice are low. Over time opinions will shift to it being more accepted and especially with all the successful implementation shown before.

"it's resemblance to commercial gaming platforms may lead some to perceive VR platforms as lacking credible and validated training benefit". To overcome this VR needs to show a real-world difference, to show its value in the business world.

Next "preliminary training is also required so users can effectively use new systems" this is something that once again will go away with time, but in the meantime, you need to give most users of these devices training in how to correctly operate the hardware as most people have not had the chance to.

Lastly "compared to real-life exercises, simulated scenarios still lack the direct hands-on experience and face-to-face interactions that real-life exercises provide" this is an issue that will always be prevalent, VR is as the name implies virtual and its ability reproduce reality is always going to be the hardest challenge to overcome.

What has been learned is that VR is a flexible and more effective means of doing training in many businesses. The implementation will lead to not just increased profits through reduced costs but also higher quality safer training. The project hopes to make VR training simulations easier to create, to allow more people to have access to all these benefits.

3.1.2 Health and Safety in VR

Health and safety is a slow-moving field; everything is done by the book and innovation is slow. These values are essential as health and safety should not be exciting; it needs to be informative and absorbing. However, it has come to a point to where something needs to change, people are becoming more easily bored, and the same old lectures and rules are no longer enough to get the message across to people.

VR allows for a new and exciting way for users to learn essential lessons. For example, there was a trial done with children using VR to teach them proper pedestrian crossing procedure. Half the sample used the Software, and the other half did not. The results from this were that "Learning, identified as improved street-crossing behaviour, transferred to real-world behaviour" (McComas, et al., 2002).

3.1.3 The online training VR environment

EON Reality is a company who are working towards improving VR and education; they have created a VR application which is similar to what this project aims to create. They call it Creator AVR it is a software which allows teachers to create VR environments with tasks and objectives based on models in that area. After the teacher has created the environment, then the student can attempt to complete the tasks put forth. The software uses highly detailed models with which the user interacts. Below is a picture showing the interface that teachers can use to create their environments.

Figure 4 Picture of Creator-AVR (EON Reality, 2019)

3.1.4 Movement in VR

Movement in VR is the ability to move the user around the world, movement is what causes a lot of the motion sickness problems when in VR, this means that doing it in the right way is of the utmost importance if the user has an enjoyable experience in the application. There are many types of movement that are possible in VR which this section will cover.

3.1.4.1 Walking

Walking is the most common movement method when it comes to VR. The most integral part of VR is immersion, and nothing is as realistic or intuitive as walking, this means that tutorials most likely will not be required. "this mimics real-world locomotion and therefore provides a high degree of interaction fidelity. Walking enhances presence and ease of navigation" (Usoh, et al., 1999). Walking may not be the best choice if extensive distance travel is required as this may not be feasible for the user if tethered to the headset, but in small to medium size environments, this works well. The other issue with walking is that users may become tired from walking which will reduce the amount of time playing.

One example of implementation is to have the user walk around a room with the headset on, this movement matches the view within the application. Because the visual cues within the headset match, the vestibular cues of the user moving around this causes a minimal amount of motion sickness, this method of implantation limits the users physical space to play in. A technique called Redirected Walking can be added it "is a technique that allows users to walk in a VR space larger than the physically tracked space. This is accomplished by rotation and translation gains that are different from the true motion of the user, directing the user away from the edges of tracked space (or physical obstacles)" (Razzaque, et al., 2005).

An option if space is an issue is the ability to walk in place; this has the user making walking gestures to cause the player within the application to move. This method is like the human joystick approach which has the user acts as a joystick and move their body in a direction to translate to movement.

Lastly, we have treadmills, which are becoming more commonly used for these VR experiences especially omnidirectional ones. What this means is that they can move in every direction allowing 360degree of motion. The treadmill is not as immersive as walking around in an environment, but they come with the main upside of being able to have an unlimited distance that can be covered. Which means there would no longer be a cap on the space that a VR environment can be.

3.1.4.2 Steering

Head steering is a way of moving the user in the application by them moving their head in some way; This allows movement over long distances without physical exertion. Movement is not just limited to just the head; it can be other body parts or even controllers. As learned in the section 3.2.2 since the user will not be physically moving in any of these scenarios techniques need to be used to reduce motion sickness as there will be a conflict between the visual and vestibular cues.

The first example of this technique is one seen a lot due to its simplicity. It is known as navigation by leaning and is when the user moves their body left and right to infer a change in the direction of the player in the application. These require no additional tracking except for the head. Motion sickness has the potential to be an issue if visual acceleration is a part of the movement tech.

Next, we have a movement based on the gaze of the user; this is where the player will move in the direction that the user is looking. An external controller can control motion speed. This technique is simple for the user to pick up but can cause issues if the user looks around when moving as unexpected changes in acceleration and rotation could cause sickness. Next, we have torso-based steering also known as tank based steering. It is where a controller is used to control the movement of the user, but unlike gaze-based steering, the movement of the stick is not relative to the way the user is looking but is constant. Like a tank, the user can swivel their head but forward will always be forward. This technique has a higher chance to cause motion sickness for users as it is a lot less intuitive for people to pick up. Dual analogue steering has been a staple for movement, "This mapping is surprisingly intuitive and is consistent with traditional first-person video games" World grounded steering devices like flight sticks, and steering wheels work well as control method since they add to the immersion levels of the experience for example if the user is playing a racing simulator. They can also have virtual steering devices in which the steering wheel is simulated using the controllers of the headset.

3.1.4.3 Automated Movement

Automated movement is a movement where the active part is taken away from the user, and instead, the application does the movement for the user. This type of movement can be done when the user is a passenger like someone on a rollercoaster or when the creator does not want to give the user the ability to explore the environment freely.

Examples of this are the roller-coaster where the user is driven along a track. Because of the creator-controlled way, this is implemented the user can add lots of the motion sickness reducing techniques like indicators of movement, constant velocity and rest-frame cues. Another common technique is target based travel where the user will point out a location in the environment, and then the camera will move over to that location potentially mimicking the frequent movement that would be associated with that translation. Lastly there is teleportation which, similarly to target based, the user selects a location in the environment but instead, they are instantaneously transported to that location. If implemented with a fade out/fade in of the environment when teleported this technique leads to a low chance of motion sickness. "Unfortunately, straightforward teleportation comes at the cost of decreasing spatial orientation—users find it difficult to get their bearings when transported to a new location" (Bowman & Hodges, 1997).

3.1.5 Map editors

In this section, is a brief history of the modding scene and map editors. From that, some examples of excellent map editors of many different types will be discussed. "The term mod is derived from the act of *modifying a game*. To mod a game is to create custom levels, objects, characters, or even unique or stand-alone game from an existing game engine" (Beal, 2019) and a map editor is "A level editor (also known as a map, campaign or scenario editor) is software used to design levels, maps, campaigns, etc. and virtual worlds for a video game"

3.1.5.1 History of Modding and Map Editors

"The original mods can be traced as far back as the early '80s with a total conversion called Castle Smurfenstein being widely considered the first. This mod was a parody of Castle Wolfenstein" (modmoddermodding, 2011). all the enemies in the game were replaced with Smurfs, all the audio was Smurfs and new music. They used the sector editor for the game engine to change all this and create something new. Mods of this nature would continue to be released and developed on, some become games in their own right.

Figure 5 DOTA mod for Warcraft 3

Mods are slowly becoming less popular, in their place developers are putting Map editors into their games. These allow the user to create their own levels or maps for a game, upload them to the internet for others to play, rate and review them. These systems have seen to add longevity to these games as there is always new content being made for them. the next section looks at many different examples of this being used.

3.1.5.2 G.E.C.K (Garden of Eden Creation Kit)

In 2008 the video game Fallout 3 was released, made by Bethesda Softworks. It was a first-person RPG. Several months after its released Bethesda released the Garden of Eden Creation Kit named after an item in the video game. This is an external program that could be downloaded for free; it gave the user the same tools that the developers used to make the game. This gave users the power to do anything in the world as they had all the options; nothing was held back. Because of this, mods are still coming out for both this and the game sequel Fallout: New Vegas.

The tool itself is very complex and has many options. It works like a traditional windows programs with menus and dropdowns, to even get started there is a 10-part tutorial on the website and a wiki filled with articles detailing how to use the tool. menus detail everything that the user would want to add to the objects. Preview screens are used to show the current work on the objects. The tools use a 3D example of the original outcome to show what the creation would look like.

Once the user has completed their creation, they can then load it into their game by using a separate program called a mod manager. The mod manager is essential due to the complexity of the mods that can be installed there is a high chance of instability within the game if the mods are loaded in the wrong order or if they are incompatible with each other. If a user wants to share their creation with others, the traditional solution is to upload them onto a third-party website. The user can add descriptions and info to interest other uses in downloading their creation.

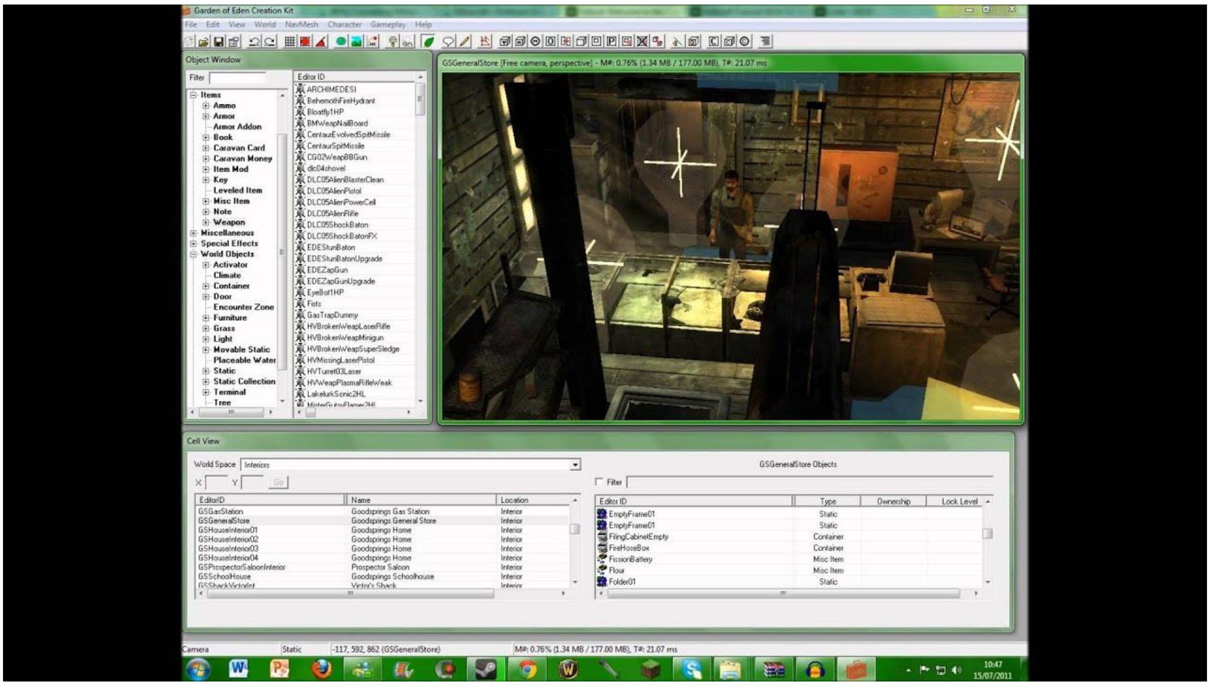


Figure 6 G.E.C.K Creation Environment

3.1.5.3 Halo Forge

"Forge is a game mode originally released in Halo 3 designed by Bungie to allow players to customise, save, and share maps for Custom Games" (Halo Alpha, 2019). Forge was not a onetime thing in the Halo series as it was incorporated in many of the sequels to the game. In the forge, the user can change the look of maps, the spawn points and objectives. They move around the map placing objects around the place as they please.

In addition to moving objects around the map, the user can change the properties of the objects such as respawning. The maps have item limits that stop the user from placing too many objects in a map; this was done for stability reasons. Another mechanic is that the user can add filters that go on top of the whole map making it look a certain way; these are pickable from a list.

Once they have made a map, it is saved to their account, and they can easily create a game and invite their friends to play it with them. The game mod can also be uploaded to the internet where other players can download their creation. All of this is done within the game, no external tools.

Figure 7 Halo Forge Creation Environment

3.1.5.4 Super Mario Maker

"Super Mario Maker is a video game which allows players to create their own levels from the Super Mario series, and then publish those courses to the Internet for other players to experience." (wikipedia, 2019). The game was made by Nintendo in 2015 to act as a celebration of the 30th anniversary of the original Super Mario Bros.

The games main feature is the ability to create the users own Super Mario levels with a map editor. The game uses a grid system as a base for the creation of levels. On this grid, the user can place objects from the games. They can place objects on top of each in the grid and break traditional rules of Mario games. They also can add sound effects even allowing users to create their own sound effects by using their microphone. There are six different course themes which are the background for the environment that make the maps the users on. There is also the ability to change the game that the course is based on from multiple different types from Mario's history. A limit on the game is that the courses the user makes need to be beatable as if they are not then they cannot share them with others.

The game has plenty of options for finding new courses through filters and search queries. So, when users have created their course, they can upload it to the marketplace of sorts where others can download and play their course.

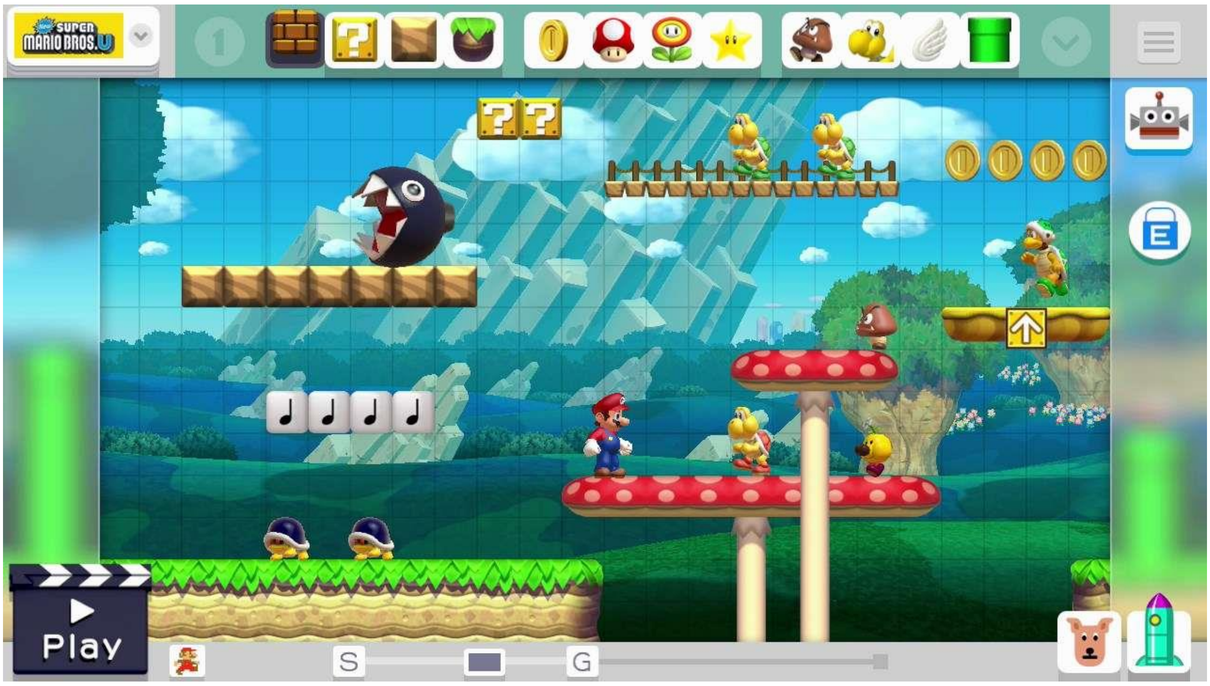


Figure 8 Super Mario Maker Creation Environment

3.1.5.5 Line Rider

Figure 9 Line Rider Creation Enviroment

3.1.5.6 *Beat Saber*

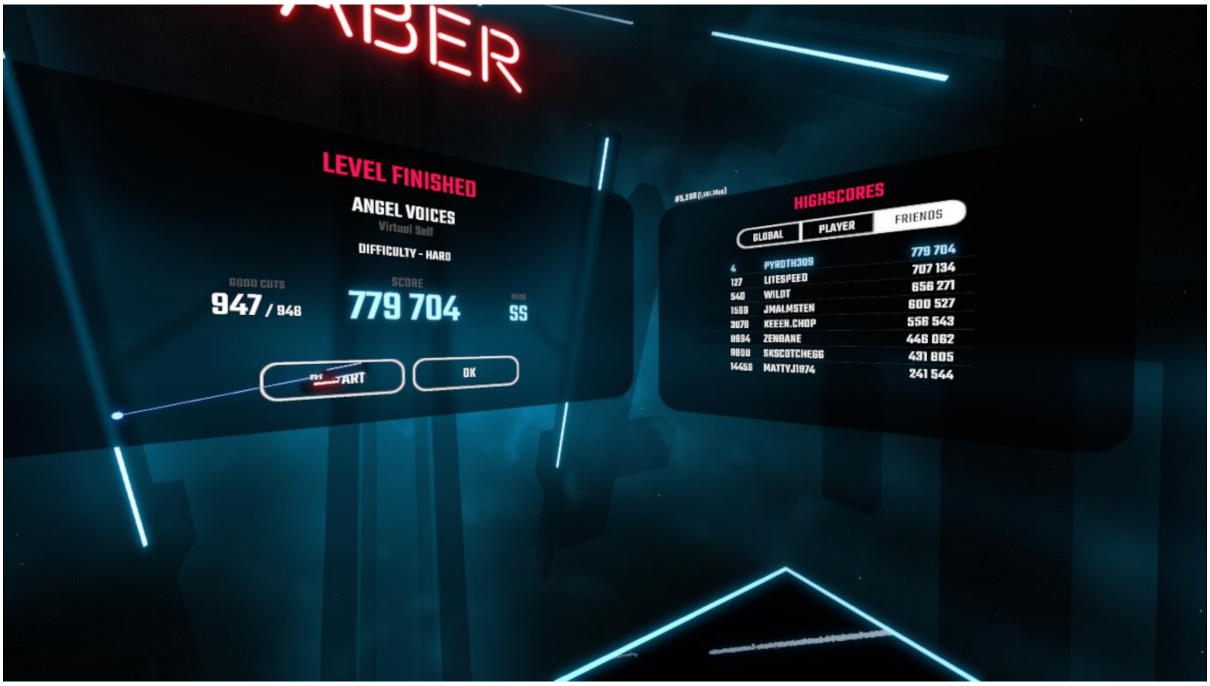


Figure 10 Beat Saber Menu

3.1.6 What The Health and Safety Department want?

the customer for the project is the Health and Safety department at Loughborough University. They have provided an outline of what they want as an output for the project(Appendix 'Number') This consists of a storyboard of what they want the VR experience creatable with the project to be. With that in mind, the storyboard is used as a checklist of features that the authoring tool should be able to do. In this section, the storyboard will be analysed to make a list of features that need to be included.

The first feature that needs to be implemented is assets that are comparable to the ones in Hazelrig building; whether these are exact copies or assets that are similar. Being able to add new assets into the project flexibly is an essential feature as it will allow for the authoring tool to be used for more and more scenarios.

NPC character is a feature that is requested as well; the implementation of NPC is quite hard unless their coding is preconfigured. What this means is letting the user control the movement and actions of an NPC will add complexity to the project but having NPC that are preconfigured to do specific actions or move in a certain way is a lot easier to implement.

The next feature is sound which is an integral part of the VR experience especially if immersion is a factor. Sound will be similar adding objects to the authoring tool.

They ask to have a set of tasks that the user must do and a specific order to them; this adds a level of complexity to the authoring tool but is possible. The specific order part will be the tricky part to implement.

Movement is something that is requested, all the different types of movement that can be added to VR and the most applicable one will be added had been discussed in section 3.1.4.

Interaction with objects is a feature that is wanted; this means that some form of input must be added to the application to allow for this to happen.

Choices have been asked for; this feature will be difficult to implement due to the limited nature of the authoring tool. The ability to implement this will be tied to how tasks are added to the tool and whether being able to have diverging paths is possible.

Hand interaction has been requested the ability to implement this is based on how interaction is done within the application.

The ability to implement all the features that are asked for is all dependent on how the authoring tool is created and its limitations.

3.2 Optional Background Research

In this section are some additional research that was done into many different areas of VR, these do not directly affect the project but more indirectly effect as they are about the world of VR.

3.2.1 Limitations in VR

Virtual reality has some limitations not just in the hardware but also in the software; in this section, we will go over some of the issues that have been found using a book called *Human-centered design for virtual reality* as a starting place for some findings.

3.2.1.1 Physical Fatigue

First is Physical Fatigue of the user which can be caused by "weight of worn equipment, holding unnatural poses and navigation techniques that require physical motion" (Jerald, 2015). In the early day of VR the headsets used to weigh a lot, as seen in **Error! Reference source not found.**, but these days the weight has decreased reducing the strain on the user. An important part of the correct construction of these headsets is that the "centre of mass is far from the head's centre of mass" (Jerald, 2015) If this is the case, the user may have to exert extra force to make sure they are balanced which can increase physical fatigue.

Figure 11 Virtual Boy Headset

The fatigue for the user does not end there, concerning the arms of the user, controllers have become a lot lighter. What is an issue is what is known as "Gorilla Arm is arm fatigue resulting from extended use of gestural features without resting" (Jerald, 2015) Thus, it is essential not to have interfaces that require the user to have their arms in front of them for any longer than a few seconds or else discomfort will ensue.

"Navigation via walking can be exhausting after a period of time, especially when users are expected to walk long distances" (Jerald, 2015). Even standing can be exhausting for some users let alone walking so finding ways to accommodate these cases is essential.

Still, on physical fatigue looking at the headset fit on the user, this means "how well a headset fits a user's head and how comfortable it feels" (Jerald, 2015). Headsets can cause discomfort for the user due to the pressure on the head caused to get the headset to fit correctly; these issues are more prevalent if the user wears eyeglasses as there is another object in the way to making a good seal. Most headsets solve the problems with discomfort by having the ability to be adjusted and tinkered until the headset is in the perfect position "such adjustments do not, however, guarantee quality fit for all users". Getting the headset to fit tightly is vital for two reasons, the first being that if the headset is not on correctly then the headset screens will look blurry and the user will not have an enjoyable experience when using the software. The second is that with the amount of body and head movement the chances of the headset falling off is high if it has not been correctly put on.

3.2.1.2 *Physical Trauma*

Physical trauma is possible when using VR; this "is an injury resulting from real-world physical object that is an increased risk with VR due to being blind and deaf to the real world". To reduce injuries of this nature, the first thing to do would be to allow for sitting which stops hazards like falling or tripping over as the user is in place. Another way to reduce issues is by having a human spotter who assists the user in making sure that nothing goes wrong as they are using the headset, they can also carry the wires to make sure they do not trip over. Repetitive strain injuries can be caused by "using rapid carpal and metacarpal movements … any interaction techniques that requires continual repetitive movements are undesirable."

3.2.1.3 *Hygiene*

Next on to one of the unique limitations of the hardware, that being the hygiene of the headsets. "VR Hardware is fomite – an inanimate physical object that is able to harbour pathogenic organisms (e.g, bacteria, viruses, fungi) that can serve as an agent for transmitting diseases between different users of the same equipment" the thing that makes this issue unique is the location of the peripherals as unlike keyboard or mice the interaction is on your face. The face produces oils and sweat, add the heat of the device and if the experience is of an active variety that means even more sweat onto the headset. Most traditional cleaning methods like washing machines or rubbing alcohol do not work as well due to the nature of the headsets. The solution is removable pads that are between where the users face and the headset, and these are discarded after each use to stop any transfer of undesirable diseases a picture of which is **Error! Reference source not found.**. The controllers can also be a disease-ridden hazard, so the use of hand sanitizer is advised.

Figure 12 HTC Vive Disposable Hygiene Cover

3.2.1.4 Latency

The most crucial task of VR is to trick the user into thinking they are in a different world, one of the limitations stopping that is Latency. “Latency is the time a system takes to respond to a user’s action, the true time from the start of the movement to the time a pixel resulting from that movement responds” the consequences of the users perceiving or feeling the effects of latency is that the scene will appear unstable and incorrect to the user. “VR Latency is a major contributor to motion sickness” an example of latency is degraded vision this is seen if the user were to move their head and then stop if there is latency the scene will keep spinning even though the user has stopped. These issues cause the user to have a break in immersion and remember that they are in a simulation.

The causes of this latency are broken down into the term System delay which is “the sum of delays from tracking, application, rendering, display and synchronisation among components”. Firstly, tracking delay which “is the time from when the tracked part of the body moves until movement information from the tracker's sensors is input into the application.” The way tracking depends on the type of headset in use. All the potential VR headsets go into two categories, either tracking is done by the headset or by external sensors. The pros and cons are straightforward if the tracking is on the headset the tracking delay will be less because the network delay of transferring that data to the computer running the software is not relevant. The reason external sensors are in use is that it allows the software to gather a lot more information about the user. The most common reason for an external sensor is that they allow for movement of the user in the real world which translates to the virtual one.

Secondly, there is application delay which is “the time from when tracking data is received until the time it is passed onto the rendering stage” this consists of updating the user’s position within the game world, game physics and event running. The delay changes wildly depending on the complexity of the game world that in use.

Thirdly, we have the rendering delay which is “the time from when new data enters the graphics pipeline to the time a new frame is resulting from the data is completely drawn”. Like the previous type of delay, this is highly dependent on the complexity of the software, whereas the previous delays come from the programming complexity of the software, rendering delays come from the graphical complexity of the software. These two together are the lead contributors to the Frame Rate of the system which is the number of times the system renders the scene per second. “Fortunately, rendering delay is what content creators and developers have the most control over”.

Fourthly is the Display delay which is “the time from when a signal leaves the graphics card to the time a pixel change”. Lots of different types of screens are used to display the data on, the most common being LCD and OLED these influence the display delay. Many factors attribute to Display delay the first being the refresh rate of the display which is the “number of times per second that the display hardware scans out a full image” the relationship between Frame rate and refresh rate is essential. For example, if the graphical pipeline is outputting at 30 frames per second and a screen with a refresh rate of 120hz is in use, the user will only see the data at 30 fps and will most likely encounter latency issues. The reverse is also true if the graphical pipeline is outputting at 120 FPS and the screen has a refresh rate of 60hz. The user will only see the display at 60 FPS.

Double buffering is a solution to the issue of when reading and writing of the same data occurs. This occurrence will happen when the graphical pipeline is rendering a frame at the same time the display is outputting the next frame to the display; this will cause incomplete frames to be displayed causing loss of immersion to the user. What double buffering does is "The display processor renders to one buffer while the refresh controller feeds data to the display from an alternate buffer" this is where vertical sync comes in which allows there to be a consistency to when swapping the buffers. If there is no vertical sync used Tearing can occur which is "during viewpoint or object motion and appears as a spatially discontinuous image, due to two or more frames contributing to the same displayed image". With V-Sync the displayed image is only ever from one rendered frame. Response time "is the time it takes for a pixel to reach its intended intensity" this value changes depending on the technology of the display in use. Persistence is the amount of time a pixel remains on the screen before going away. Once again, this changes on the technology of the display; if this value is low, ghosting can happen which is where pixels can be seen in later frames when they are not supposed to be.

Lastly on the set of delay types we have synchronization delay which is as "the delay that occurs due to integration of pipelined components" this is made up of mainly waiting for signals before moving onto the next stage, as later components of the pipeline are waiting for data from previous ones before they can do any processing.

3.2.1.5 Eye Strain

Onto eye strain, one of the main attributes that cause eye strain is the Accommodation-Vergence Conflict which is where "In the real world, accommodation is closely associated with vergence, resulting in a clear vision of closely located objects. If you focus on a near object, the eyes automatically converge, and if you focus on a distant object, they diverge" there becomes an issue in VR as the "relationship between accommodation and vergence not being consistent with what occurs in the real world" this causes eye fatigue and discomfort for the user.

Next is the Binocular-Occlusion Conflict which occurs when "occlusion cues do not match binocular cues, e.g. when text is visible but appears at a distance behind a closer opaque object" this will confuse the user and can cause uncomforting feelings.

When using VR, some people encounter VR Aftereffects which is "any adverse effect that occurs after VR usage but was not present during VR usage". There have been examples of perceptual instability of the world, disorientation and flashbacks. "about 10% of those using simulators experience negative aftereffects" there is a connection between people who experience the most motion sickness and the aftereffects. "VR entertainment centres require that users not drive for at least 30-45 minutes after exposure."

3.2.1.6 Lessons Learned

Looking into the limitation of VR shows some lessons that are now into the project. The first being that VR has a physical toll on the user, this will be stopped by making sure the experience is viable standing and sitting, making sure the user does not get gorilla arms by having a well-made interface that makes sure that is not an issue. The user will not walk around a lot as that causes discomfort for users; the movement of the user is essential through which will need much research to decide the correct approach. Headset choice is vital for many reasons the first being the ability for the headset to comfortable fit onto a person's skull as inflexibility here can cause for users to not be able to use it. The headset choice is crucial as it will profoundly affect the Latency that will happen due to each one having different sensors, computers and displays. Lastly knowing the full pipeline of VR shows how important the programming and graphical choices are going to be when making the application, as they can cause substantial latency issues if done incorrectly.

3.2.2 Motion sickness in VR

In this section is research into Motion and general Sickness in VR, the reasons for and the effects. To start Motion Sickness is "adverse symptoms and readily observable signs that are associated with exposure to real or apparent motion" (Lawson, 2014). The part of the definition that is related to VR is apparent motion which is what causes the issue that is associated with VR. The symptoms of motion sickness are "general discomfort, nausea, dizziness, headaches, disorientation, vertigo, drowsiness, pallor, sweating, and, in the occasional worst case, vomiting" (Kolasinski, 1995). Splitting Motion sickness, into either visually induced motion, and physically induced motion, visually induced motion is the type that is associated with VR and is what we are trying to reduce.

Movement in VR is significant to allow more immersive and enjoyable experiences but to have movement we need Scene Motion which is "visual motion of the entire virtual environment that would not normally occur in the real world" (Jerald, 2015). There is intentional scene motion and unintentional scene motion; the difference is whether the scene motion was meant to happen or not by the code. Both, if noticeable can causes motion sickness. Another keyword is Vection which is "an illusion of self-motion when one is not physically moving in the perceived manner" vection can be another cause of motion sickness if done incorrectly. There are no concrete facts on what causes motion sickness to occur only theories here are some of them.

3.2.2.1 Theories of motion sickness

The first and most accepted theory is the Sensory Conflict Theory it states that motion sickness "may result when the environment is altered in such a way that incoming information across sensory modalities (primarily visual and vestibular) are not compatible with each other and do not match our mental model of expectations." (Reason & Brand, 1975). This theory says that seeing things that do not agree with what the body of the subject is feeling causes the motion sickness to occur.

The next theory is known as the Evolutionary Theory which offers a reason for why we feel sick when inconsistency occurs. It says that "it is critical for our survival to properly perceive our body's motion and the motion of the world around us" if the data from our sensors do not line up there must be something wrong, so we feel sick to tell us to stop doing whatever is causing the inconsistency. The body acts similarly to when someone is intoxicated and does the same routine to protect the body by "discouraging movement, ejecting the poison via sweating and vomiting, and causing nausea and malaise."

The next theory is the Postural Instability Theory which aims to be better than the Sensory conflict theory by being able to say when and how severe sickness will be which the sensory theory cannot do. The theory states that "sickness results when an animal lacks or has not yet learned strategies for maintaining postural stability" (Riccio & Stoffregen, 1991). The theory says that animals become sick when they are in circumstances when they do not know how to maintain balance. The theory uses the example of "sea legs" of being able to better manage seasickness by subconsciously learning strategies to control their balance aboard a ship. The theory says that since people are not used to the VR and that over time will be able to keep posture better and feel less motion sick.

Next, we have another theory that conflicts with the sensory conflict theory, one of the issues with that theory are that there are times where its expected that motion sickness to occur, but none does. The Rest-Frame Hypothesis says that "does not arise from conflicting orientation and motion cues directly, but rather from conflicting stationary frames of reference implied by those cues" (Prothero & Parker, 2003). The theory says that the brain has a mental image of all the stationary objects in the scene. The brain then considers all other moving objects relative to the stationary ones. Traditional stationary objects are rooms or backgrounds. When sensory cues come that do not match up with the brain's current idea of how the room should move then motion sickness occurs. This idea means if there is a scene motion that causes sensory conflict. It will not cause motion sickness if it is possible as seen in the rest frame of the room. Therefore motion sickness only sometimes occurs. So, to reduce motion sickness, it is important to make the rest frame of the scene consistent so that all movement is relative towards it. Proof that this theory has some validity is the fact that "Motion sickness is much less of an issue for augmented reality optical-see-through HMDs (but not video-see-through HMDS) because users can directly see the real world, which acts as a rest frame consistent with vestibular cues."

The final theory on motion sickness is the Eye Movement Theory; this states that it "occurs due to the abnormal eye motion required to keep the scene's image stable on the retina. If the image moves differently than expected, such as often occurs in VR, then a conflict occurs between what the eyes expect and what occurs. The eyes then must move differently than they do in the real world in order to stabilise the image on the retina. As a result of this discrepancy, motion sickness results".

3.2.2.2 Causes of motion sickness

Onto, the actual causes of motion sickness to occur not the theories as to why. These reasons are heavily linked and interlocked with many of the theories. The first defined set is system factors; these are causes that are related to the hardware and technological shortcoming of virtual reality.

Latency, as discussed in section **Error! Reference source not found.**, is one of the leading reasons for sickness as the delay between movement and change within the VR experience causes the issue. Calibration of the hardware if poorly done will cause the users VR experience to degrade, poor calibration can cause "inappropriate scene motion/scene instability that occurs when the user moves her head" if scene motion happens without the user expecting it to happen this will cause sickness. Tracking accuracy is very similar to Calibration; just the error is in a different section of the pipeline. Like calibration, poor tracking accuracy can cause inappropriate scene motion to happen. If poor tracking happens to the hands, this does not cause motion sickness but does cause the user to lose immersion. Tracking precision is different from accuracy; accuracy is how close to the correct answer the instrument has achieved, where precision is to what degree the instrument can detect. Low precision will cause the user to jitter which is where slight movements happen to the user when they are standing still once again causing unintentional scene motion. Next, Lack of position tracking, if the ability to track the position of the user, is not available this can cause motion sickness in a situation when the scene needs to move but does not. For example, when the user tries to pick something off the ground, if the scene does not correctly move the camera to a lower level, this will cause sickness. The field of view of the display is vital as If it is too high the user will be more sensitive to vection. The refresh rate, display response time and persistence as discussed earlier will cause latency issues. The vergence-accommodation conflict will cause sickness if done poorly. Real world peripheral vision If the headset has been put on poorly or is of a weak construction the user will be able to see the real world and use it as a rest frame for the scene which will cause sickness as it is not a part of the virtual scene so that the brain will get confused. Headset fit, if the headset is too tight or too loose, this will cause pain or inability to focus on the content both can cause problems. If the weight of the headset is too high, this will cause the user's neck to be in pain and can lead to headaches.

This section will talk about more personal reasons why people may encounter motion sickness when they use VR. The issue here is that the "range of vulnerability varies by a factor of 10,000 to 1. Some people immediately exhibit all the signs of VR sickness within moments of using a VR system, others exhibit only a few signs after some period of use, and some exhibit no signs of sickness after extended use for all but the most extreme conditions" (Lackner, 2014).

On to the more specific personal causes, The first is a prior history of motion sickness, if the user has a history of getting travel sick, sea sick or motion sick then the chance is high that the same thing will happen if they use VR. Next, is the current health of the user, if the user is currently ill or intoxicated VR will only make them feel worse as it will increase the ill feelings. That is not to mention the hygiene or damage issue that may come with such encounters. Next VR experience, the more a user uses VR, the more the body adapts and becomes used to it, and the chances of motion sickness occurring go down. Surprisingly gender affects "Females are as much as three times more susceptible to VR sickness than males" (Stanney, et al., 2006). The explanation is hormonal and field of view differences that cause sickness to be more prominent. Age is also a factor in susceptibility, "VR sickness, however, increases with age" (Brooks, et al., 2010).

The final set of causes are ones that have to do with the design of the application. First is the Frame rate of the application as talked about early this is a big proponent of the total latency of the system which will comprise the system if poor. If the user does not have control of their movements, they can less anticipate the movements that are occurring to them and are more likely to have motion sickness. Physical head motion, the more head motion the user must do the more likely the user is to have motion sickness, so it is important to keep head motion to a minimum when possible. Duration matters like head motion the more, the worse it gets so keeping experiences short helps mitigate that. Vection as mentioned earlier can cause motion sickness, so the less of it is, the better. Same with binocular-occlusion conflict and gorilla arms. As talked about In the rest frame theory from section **Error! Reference source not found.** having rest frames in the application will reduce motion sickness. Whether the user is standing or sitting influences motion sickness; if the user is sitting the chance of sickness occurring is a less then if the user is standing this may correlate with the postural instability theory.

3.2.2.3 Reduction of motion sickness

We have seen the causes of motion sickness; in this section, going to go through some techniques that will reduce motion sickness in the project.

The first of these are real-world stabilised cues, examples like travel sickness in a car. If someone looks at a stationary object when travelling in a car, they will start to feel sick as there is a conflict between their visual cues of movement and their vestibular cues. In this case, their visual cues are saying there is no movement as they are looking at a stationary object, but their vestibular cues are saying that the car is moving. In VR the opposite happens as their visual cues are saying that they are moving, but their vestibular are saying no movement. In real life, we can stop this sickness by instead of looking at an object within the car to look at the outside of the car to see the movement of the horizon, so the cues are no longer in conflict. We can apply this inversely to VR by adding static objects to the environment, which act as a rest frame for the user to subconsciously tell them they are not moving, this reduces motion sickness to occur in the user.

Leading indicators are cues in the application that indicate to the upcoming user movement. These allow the user to prepare themselves for this movement and are thus less likely to get motion sick. "For example, adding an indicator object that leads a passive user by 500 ms along a motion trajectory of a virtual scene (similar to a driver following another car when driving) alleviates motion sickness" (Lin, et al., 2004)

Minimising Visual Accelerations and Rotations is another way to reduce motion sickness; linear motion does not seem to have a significant effect on the user. However, any visual accelerations appear to be a problem for users; the same goes for rotation which is a form of acceleration. If visual acceleration is to occur there are two ways to minimise its effects on the user. The first is to have the acceleration happen as quickly as possible; the second is to have the user control the acceleration.

A researcher called Denny Unger "has found that discrete virtual turns (e.g., instantaneous turns of 30°) he calls ratcheting result in far fewer reports of sickness compared to smooth virtual turns" this appears to result in breaks in presence but does reduce the sickness. It also only works with 30° turns.

Motion platforms "is a hardware device that moves the entire body resulting in the sense of physical motion and gravity. Such motions can help to convey a sense of orientation, vibration, acceleration, and jerking" this if applied correctly to VR can reduce the motion sickness by adding vestibular cues that line up correctly with the visual cues that the application is giving the user. These need to be correctly applied as if the cues do not line up then motion sickness will occur.

Warning grids are used to stop the users from moving outside of the safe zone that the user created. These are mainly used to stop the user from damaging the real world by slamming into nearby objects. They also stop motion sickness by stopping the user from moving into areas which have inadequate sensor coverage. As stated earlier inadequate sensor coverage will increase latency and thus motion sickness.

3.2.3 VR Immersion

The main objective of VR is to be taken to a brand-new world; whether that world is a copy of the real world or somewhere entirely new, the purpose is the same to trick you into thinking you are not where you physically are. In this section, is some of the critical tactics and mechanics that are used to immerse the user some of which will relate to previous sections.

3.2.3.1 Immersion

Immersion is "the objective degree to which a VR system and application projects stimuli onto the sensory receptors of users in an extensive way, matching, surrounding, vivid, interactive, and plot informing" (Slater & Wilbur, 1997). This definition can be broken down a lot more like a lot of the terms are not apparent.

Extensive means the number of sensors that are stimulated, the most apparent sensors is visual and audio both of which is minimum for any VR experience, there are also some more advanced sensors like the physical ability to pick up controllers, the haptic feedback of those controllers and lastly the forced movement of using motion platforms.

Matching means that the audio and visual and more of the application match and make sense together if they are misaligned, that can cause a massive break in immersion.

Surroundness is the ability for the application to be a full 360 experience; This can be achieved by having a wide field of view for the user, an environment that is in every direction and finally as talked about in section **Error! Reference source not found.** having spatialized audio.

Vividness is the quality of the assets used in the application; this does not necessarily mean how realistic they look but more how correct they are in the universe of the environment. However, a big part is the quality of the assets in terms of resolution, realistic lighting methods, a stable frame rate and quality of the audio. If the game looks and sounds and plays terrible, then the user will not have the chance to immerse themselves within this world as they will be too distracted by its abhorrent qualities.

Intractability is how much the user can interact and change the world; the critical quality is that the world is changing. If the user throws an object across the room, it should stay where the user threw it, not reappear to its original position. The interaction should also be, as mentioned earlier, Matching. The interaction with objects should be what the user expects it to be as incorrect matching can break immersion.

The plot is the story of the application, like movies a great plot engages a user in the world and makes them want to continue to learn more about the world. A great plot is not required, but some explanation of why the user is doing what they are tasked with is better than having nothing.

3.2.3.2 Presence

Presence is a hard concept to explain but this quote explains it best: "Whereas immersion is about the characteristics of technology, presence is an internal psychological and physiological state of the user; an awareness in the moment of being immersed in a virtual world while having a temporary amnesia or agnosia of the real world and the technical medium of the experience". Its what people mean when they say they were sucked into a book or movie, its the feeling of forgetting the user is in a room with a VR headset on. If an application is immersive, it does not mean the user will feel presence but the better the immersion of an application the more likely for there to be a presence within its world. A break in presence is when the illusion is broken like seeing how the magic trick is performed it can ruin the entire experience. A typical example is when the user feels the long lead of the HTC Vive headset. In the virtual world, there is no wire following them around, but as soon as they feel the wire scrape their leg, they are instantly no longer in the experience.

3.2.3.3 Types of illusions

Immersion is an illusion, and there are many different types of illusions that are put upon the user. The first illusion is one of being in an environment; in this case, an environment that is not one's own, This happens since what is being shown and heard is this new environment, so the user must be there. This illusion is helped by not being stuck in a single position, the more the user can move around, and everything stays consistent with how the user expects the more the illusion works. Anything that makes the world not real or fake breaks this illusion, latency problems as mentioned earlier have a significant effect on this as a user immediately senses the stuttering to be a fake and not consistent. Miscalibration again as mentioned earlier will break this as the user will not move as they expect this leads to a break in presence.

The next illusion is one of having a body, which is quite a weird statement. In real life whenever we look down, we see ourselves. In many VR application, this presence is not in them, and the user is just a flying head. The body does not need to look like one's own but it being there dramatically adds to the experience, being a character in the world, not an invisible watcher. The illusion is done correctly in most VR applications with how we see our hands in the environment and how they match to what our hands are doing in real life.

The next illusion is one of being not just being in a world or a character in a world; it is an illusion of being able to be apart of the world in the sense that they can change it and effect it. If the user picks up an object, there should be some highlight or rumble that they have touched an object in the environment.

The final illusion is one of being able to communicate in the world; this can be through a microphone in a multiplayer experience or through dialogue trees. The ability to socially exert oneself adds a lot or presence especially if characters respond believably. This illusion can be used to help users who may have a fear of public speaking by giving them the illusion of being in front of a broad audience to help them get over the fear.

3.2.3.4 Uncanny Valley

When trying to replicate, the real world some pitfalls need to be careful of; these can lead to less immersion to more. The first of which is a well-known phenomenon called the Uncanny Valley. As we slowly get closer and closer to characters that look like humans, we reach a point where instead of the characters looking realistic and empathetic instead, they look creepy and repulsive, the descent and subsequent rise are called the uncanny valley; This is best explained in the figure below

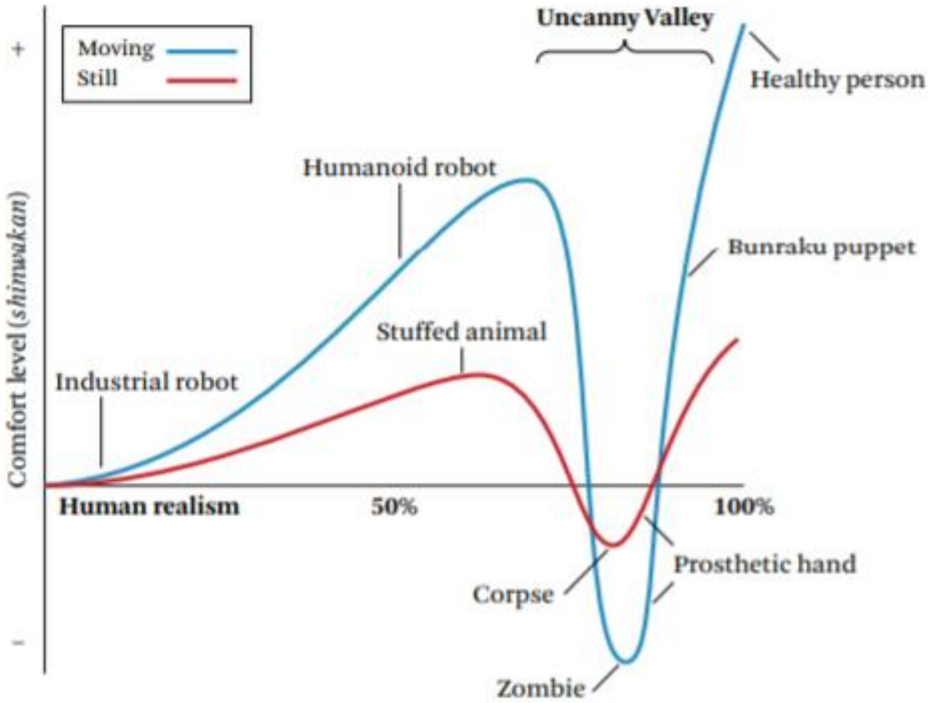


Figure 13 Uncanny Valley

3.2.4 VR Design Principles

In this section, is the techniques and tips that are used to make VR experiences the best they can be. These range from the environment to the interaction that the user has.

3.2.4.1 Environmental Design

The environment is one of the essential parts of the VR experience; it is the world that the user must believe is real. A VR environment can be split into four distinct parts. The first is the background; a great example of this is the skybox of the scene. The user has no interaction with the background; it is merely there. Next is the contextual geometry which "helps to define the environment one is in" they normally consist of landmarks that aid in wayfinding which are in the space that the user can play in. Next is basic geometries which "consists of nearby static components that add to the fundamental experience" examples of this are tables or doorways. Lastly, are interactive objects; these are dynamic items with which the user can interact. It is essential that the scaling between all these sets of objects is correct and consistent with the art style of the environment.

Colour of objects in the world is important, and this can be to imitate the likeness of real-life objects or to bring attention to a particular object with bright colours. The wrong colour can take the user out of the environment and the application. Extreme colouration or abnormal colours can be a unique selling point of new art style that grabs the user's attention. Lighting as similar to colour can be placed on an environment to exert specific emotions upon the user.

Audio is a critical part of the environment. "Auditory cues play a crucial role in everyday life as well as VR, including adding awareness of surroundings, adding emotional impact, cuing visual attention, conveying a variety of complex information without taxing the visual system, and providing unique cues that cannot be perceived through other sensory systems", since immersion is such an important objective of VR well-done audio makes a massive difference to the experience. Audio does not just have to be sound effects; it can be spoken word given instructions or story beats to the user. A great example of VR experience based heavily on quality voice acting is Accounting, which guides the user through its magical world with engaging voice acting alone.

Figure 14 Accounting

A new audio technology that has increased the quality especially in VR is spatialized audio which is "sound that is perceived to come from some location in 3D space" a great example of this although not a VR game is Overwatch, which uses audio cues to give players awareness of events happening in the map. Spatialized audio is a fantastic tool to help with wayfinding.

Wayfinding aids as already mentioned, "help people form cognitive maps and find their way in the world" they help the user not get lost in the environment. Wayfinding aids are important in VR since the ways of traversing the world as mention earlier cause the user to get lost easily. Examples are "architectural structures, markings, signposts, paths, compasses, etc." wayfinding aids that would be impossible in the real word like floating arrows can be added.

Real world content can also be added into these virtual environments if it is suited. The first type is 360 cameras which take a picture of the world in every direction from a point. Due to the nature of VR is seeing the world from every direction from a single point, 360 cameras can be easily used and added to the experience. "Instead of watching a movie through a "window," viewers of the immersive film are in any part of the scene".

3.2.4.2 User experience

Personal wayfinding aids are like previously mention aids that help the user successfully navigate the world without getting lost. Personal wayfinding aids are some which the user can interact with directly or is always on the player. The first example of this is the Map. A stable in video games for the last 20+ years, a map is "a symbolic representation of a space, where relationships between objects, areas, and themes are conveyed". Maps can be static or dynamic, static as never changing always showing the same thing and the user can use it as a reference with the virtual world. Dynamic maps update themselves with where the user is currently on them, nearby critical locations or remove fog when new areas are found. Mini-map are an extension of maps where they are always shown on the screen but in a smaller variety. Compasses are another form of this type of wayfinding. They help the user know the way they are pointing. Some compasses may add the location of markers on to them to help the user navigate more effectively like in the game. Combining wayfinding aids allows for the most effective form of location finding.

An issue that is plaguing VR content creators is not being able to control the direction that the user is looking. "Visionary VR, a start-up based in Los Angeles, splits the world around the user into different zones such as primary and secondary viewing directions" (Lang, 2015). The most important action happens in the primary direction; this is also where the user will initially be looking. The others are of less important, but information can still be conveyed in them.

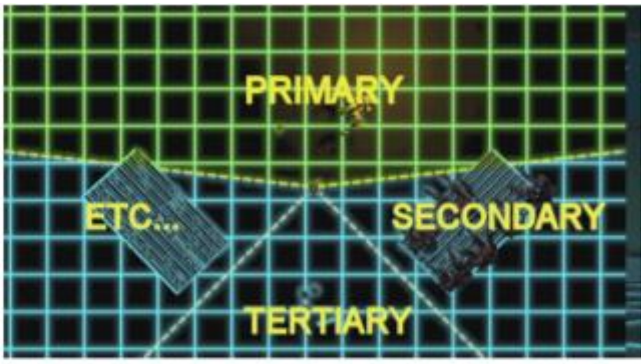


Figure 15 VR Viewing Directions

In VR we may come across other characters in the world; these can be AI controlled characters or real people. The ability to perceive other adds the question about how oneself is seen in VR. The user can be a silent protagonist, another person or an omniscient watcher. The ability to role plays as other characters have been a staple in video games for a long time. If other users are going to see the user, this means that the motion of the character must be realistic, or it will cause a break in immersion if the movement is unnatural. VR chat is a VR application that is very popular as it lets users socialise with other users in a virtual world. The avatar that the user uses to portray themselves can be anything from a virtual representation of them to Knuckles from Sonic the Hedgehog. The avatar mimics the action of the user to the other people in the application; this has brought in a brand-new form of full body communication.

Figure 16 VR Chat

3.2.4.3 Changes from traditional content creation

The most significant change from traditional content creation is a focus on the user experience as that is the whole package in VR. There are many pitfalls to be careful of, and any of them will ruin an experience. One of those pitfalls and a thing that has been mentioned before is Motion sickness; it does not matter how good the rest of the application is If it makes everyone who uses it sick then it is not very good. So, spending time to reduce the effects as much as possible is essential. Aesthetics of a game have been one of the striving points for the gaming industry to the detriment of the Frame Rate in these games. More commonly these days video games are coming out that only have 30FPS and this is fine in traditional gaming. This cannot fly in VR as anything that low will cause latency issues for the user and thus induce motion sickness. So, it is more important that the application runs well than looks good. Viewpoint is another big difference from not just other video games but media in general. The viewpoint is the human eyes, so it is essential to look and research into how humans perceive the world as this will help with adding immersion to the user.

Traditional content can be used in VR; watching movies in VR has become quite popular. There have been attempts to port non-VR games initially to VR examples, of this Skyrim and Fallout 4 VR. the challenges are making sure that the tracking of the head and hands are of high enough quality. Existing models needs to be fully realised in 3d, or else they will look unconvincing due to the depth that the user can see in VR. Traditional HUDs (Heads up display) does not translate to VR since they are usually a 2d plane overlaid on the camera, because of the lack of depth they do not look convincing in VR. The solution for this is to add HUD elements into the environment or turn them off. Camera zooming should be not done at all in VR, "A zoom lens that moves with the head causes a difference between the physical field of view and the rendered field of view, which results in an unstable virtual world and sickness when the head moves."

Figure 17 Skyrim VR

3.2.4.4 VR Input Devices

Input devises are the tools we use to interact with the virtual environment; these tools have many techniques in which they use to convey information to the application. Some techniques are only available with specific devices. In this section, we are going through the characteristics of input devices and the techniques they use to translate the user's choices to the application.

The size and shape of the controller are essential. Whether they are usable in both hands or just one if the user needs to use all their arm muscles or just a couple all of this depends on the size and shape of the controller. "Smaller devices can also decrease clutching—the releasing and grasping of an object in order to complete a task due to not being able to complete it in a single motion" (Jerald, 2015).

Degrees of freedom (DoF) "is the number of dimensions that an input device is capable of manipulating" (Jerald, 2015), buttons have1 DoF that being up/down. A mouse has 2 DoF. The hand tracking for VR should have at least 6 DoF that being a full 3D translation (up/down, left/right, forward/backwards) and rotation (roll, pitch, and yaw).

World-grounded input devices "are designed to be constrained or fixed in the real world and are most often used to interact with desktop systems". Mouse and keyboard are the most common example of this type of input device that works well on a desktop but are not very immersive or translate well to VR. The type of devices that work well in VR is those that mimic real-life controls. For example, Flight simulators sticks or Racing wheels work well. This can be used with a motion platform to create an attraction for users to at entertainment venues.

Non-tracked handheld controllers are "devices held in the hand that include buttons, joysticks/analog sticks, triggers, etc. but are not tracked in 3D space" (Jerald, 2015) these are the standard form of a controller for traditional console video games. Some VR applications do allow for the use of these controllers to be used. These controllers work better than mouse and keyboard in VR as the user movement is more flexible when using them. Some users enjoy playing non-VR games in a VR headset for the added immersion that the device gives, in these situations controllers of this variety are applicable.

Tracked handheld controllers have the same/similar functionality as a non-tracked handheld controller the differences are that these controllers can track 6DoF of movement within in them and are also tracked within the 3d space of the application. These are the most commonly used controllers for VR applications since they are tracked it can show their real-life position in the application. Add the ability for haptic feedback and the controllers can mimic many things in the VR space.

Hand-worn devices are things like gloves or Thalamic Labs Myo. "Many believe gloves to be the ultimate VR interface as they theoretically have many advantages, such as not having line-of-sight, sensor field-of-view, or lighting requirements so the hands can be held comfortably to the side or in the lap with no concern of losing track, resulting in less gorilla arm if the interaction techniques are designed well." (Jerald, 2015). The current issue is that the technology is not there yet for these to be true. The ability to accurately follow the movement of all the little finger joints is still out of reach. However, given how quickly the field of VR is evolving it will not be long before we have this.

Figure 18 Thalamic Labs Myo

Bare hands input devices via sensors aimed at the hands are the next stage after haptic gloves, but the technology in this area is more behind that of the gloves. One negative with using the hands over the gloves is the lack of feedback that the user would receive in the real world as their hands are not in contact with anything. Another issue is that the user will have to keep their hands in a specific LOS of the sensor which may cause the user to have fatigue issues. The advantage of seeing the users own hands move within the virtual world will be highly entertaining to the field.

3.3 Existing tools

In this section, is the software that is currently on the market that will help complete the project. Splitting this section into two subsections each of which correlate to a different part of the project. They are the authoring tool and the Virtual Reality application.

3.3.1 Authoring Tool

Lots of examples of authoring tools have been shown in section 3.1.5; they will be used as inspiration for creating the authoring tool for the project.

3.3.2 Virtual Reality Application

There are currently two engines that are capable of quickly making VR applications; they are Unreal Engine and Unity. In this section, both will be compared and explained.

3.3.2.1 Unity

Unity is a game engine developed by the Unity Technologies company; it was first released "in 2015 and targeted only OS X development. Since then Unity has developed and currently supports 27 platforms including VR" (Smid, 2017). Unity's best quality is its ease of use; it is straightforward for people to pick up and use. It is very quick for someone to create their project and then export it to several platforms easily. "Scripting in C# is fast and efficient" (Smid, 2017). The last point is the community that unity has built around itself. Unity is the most developed on engines especially for solo or small studios with this they have lots of places to find debugging help. This community has also created the asset store a place where users can find not just in-game assets like terrain or objects but scripting plugins that can build the whole game without the use of any coding.

3.3.2.2 Unreal

The Unreal game engine is developed by Epic Games. Its main selling points are its quality visualisation, vegetation and terrain. "Unreal has Blueprint system for visual scripting. Blueprints are graphs made of blocks connected. The connection creates certain logic instead of the scripts." (Smid, 2017). Unreal uses C++ as its scripting language, the scripting language unlike Unity's allows access to the whole system of the project; this allows finer optimisation and control over the project. The renderer used for Unreal is of high quality and has many desirable post-processing effects. On the negative Unreal has a complicated framework that is hard to learn; it is also not well suited to small projects. Lastly, it does not have as big a community as Unity so finding assets and help is more laborious.

3.4 Existing technology

In this section, the technology that is currently on the market that can be used for in the project. In this case, many types of VR headsets are currently available. This section will be talking about only "Mobile" this means that they are not tethered to a console or PC as they are the only ones that can be used for the project.

3.4.1 Mobile

Mobile VR is slowly as the years go past is getting better and better. The reason for this is that mobile processors and graphical processing units are getting more powerful; these improvements are allowing for more higher fidelity more immersive VR experiences.

3.4.1.1 Oculus GO

The Oculus GO is a headset developed by Oculus who is currently being backed by Facebook. The GO is a standalone headset; what this means is that all the hardware and software required to run VR applications is all in the headset. This means that the user does not need to plug anything into the headset it just works out of the box. The GO is very comfortable and light. It comes with speakers that make spatial sound, so headphones are not required. The headset comes with a controller that is of 6Dof; this controller allows for smooth movement through the GOs menus or to access menus and games. The headset is only 3DoF which means that it will not track any movement in the real world to the virtual world because of this its best used when sitting down. The GO has a multitude of apps and games that are available to be downloaded from the store that allows hours of content to be played.

3.4.1.2 Samsung Gear VR

The Samsung Gear VR is a mobile headset made by Samsung; the Gear VR works differently to the GO as it is not standalone. To get the headset to work, the user needs to place their Samsung phone into the headset, and from there, the headset becomes the VR display and experience. Because of this, the experience is highly dependent on the type of phone the user has, the better the phone, the better the experience. Like the GO the headset is only of 3DoF, and the device does not come with any controllers. An important thing of note is that only Samsung phones are compatible with the Gear VR so if the user has any other, they will not work. The Gear VR shares most of its VR library with the GO.

3.4.1.3 Google Daydream and Cardboard

Google has two VR Headsets; the first is the cardboard. The cardboard is the most accessible VR headset in the world; it is a piece of cardboard that the users places their phone within to use as a headset. It works with most phones but like the Gear VR the better the phone, the better the experience. The daydream is Google's more premium headset; it's more comfortable and more visually appealing. It also comes with a 3DoF controller to allow for more complex gameplay. One of the best advantages of these headsets is that any phone can be used to play them unlike the Gear VR, this allows iPhone users access to VR.

3.5 REQUIREMENTS TABLE

From all the background research, the brief and the example that health and safety have given a table of requirements can be created to make sure the project does its goals. The MoSCoW rating method will be used to rank the requirements

Requirement No	Description	Rationale	MoSCoW rating
1	The user will be able to add an	The project aims to allow people to	Must

	object on the authoring tool and see it in the VR Application	create their own VR environment; objects are required to make that environment the user's own.	
2	It is easy to add new assets to the project, to allow for continual additions	As stated in Section 3.1.6 health and safety want assets that are like their environment, so having the ability to add new asset easily is important	Should
3	The user will be able to add tasks to the authoring tool, and the user can follow those tasks in the VR application	Health and safety want a storyboard, having a story of a room necessitates a way for the user to control how the player moves through the environment thus tasks can be used to give direction to the player.	Must
4	The user will be able to add sounds into the authoring tool and hear it in the VR application	As talked about in Section **Error! Reference source not found.** sound is vitally essential to immerse the user in the experience.	Must
5	The user can move around the environment	Talked about in Section 3.1.4, movement is vital for the user to have the freedom to explore the environments created and immerse themselves in the story of the Scene.	Must
6	Interaction with objects in the environment	Health and safety would like the user to interact with eh objects that have added to the Scene.	Should

7	The user has the option to have choices on what options they can do when in the VR application	Health and safety want the option to have diverging paths based on choices that the user has made	Should
8	The user can interact with the VR environment by using their hands	Health and safety requested this feature when it comes to testing if an object is hot.	Could

4 DESIGN

In this section of the report, all the different methods for creating a solution to the problem stated at the beginning of the report will be explored. Starting with a look at the overall design of the project than a closer look at the authoring tool, database and mobile application.

4.1 SYSTEM DESIGN

The requirements of the project as stated in section 2.2 is to allow academics, who do not possess coding or game development skills, to make their VR environments easily and quickly. This can be achieved in a multitude of different ways.

4.1.1 Authoring tool and Mobile application combined

The first proposed way is to have the authoring tool and mobile application all be the same application; this is how the authoring tools are done in Super Mario Maker (section 3.1.5.4), Halo Forge (section 3.1.5.3) and Line Rider (section 3.1.5.5). The user will load up the application and will have a choice of creating or playing a scene.

There is a big difference between the examples and this project, and that is that playing the scene will be in VR, this means that the creation of the scenes will also be in VR as well since it is not possible to go out of VR in VR applications.

The advantages of having the application combined are:

- Only one application needs to be downloaded and installed.
- Ability to quickly move from creation to playing.
- Updates only need to be pushed to one application.

The disadvantages of having the application combined are

- Designing a room in VR is difficult for the user.

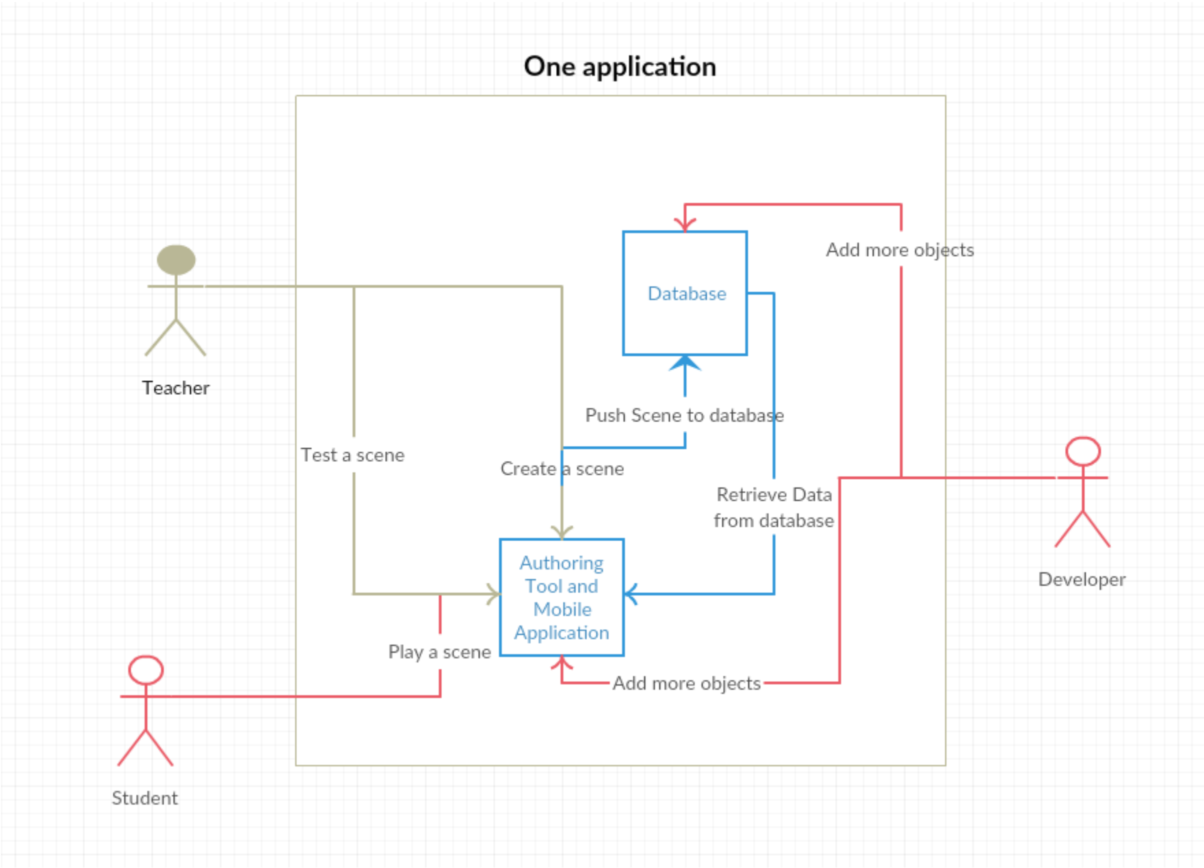


Figure 19 System diagram for one application

4.1.2 Authoring tool and Mobile application separate

The second proposed solution is to have the authoring tool and the application the users use to be separate, this is how it is done in the G.E.C.K (section 3.1.5.2) and Beat Saber (section 3.1.5.6). If the user wants to create scenes, they will use the authoring tool. To play and test the scenes then the mobile application will be used.

The advantages of having the Authoring Tool and Application separate are:

- The authoring tool can be made without having to interact with the application directly.
- Ease of use can be a focus of the authoring tool.
- Students would only interact with the mobile application.

The disadvantages of this would be:

- Testing creations would require changing to the application.
- Might be hard to replicate the final product in the authoring tool.
- Updates need to be made to three places.

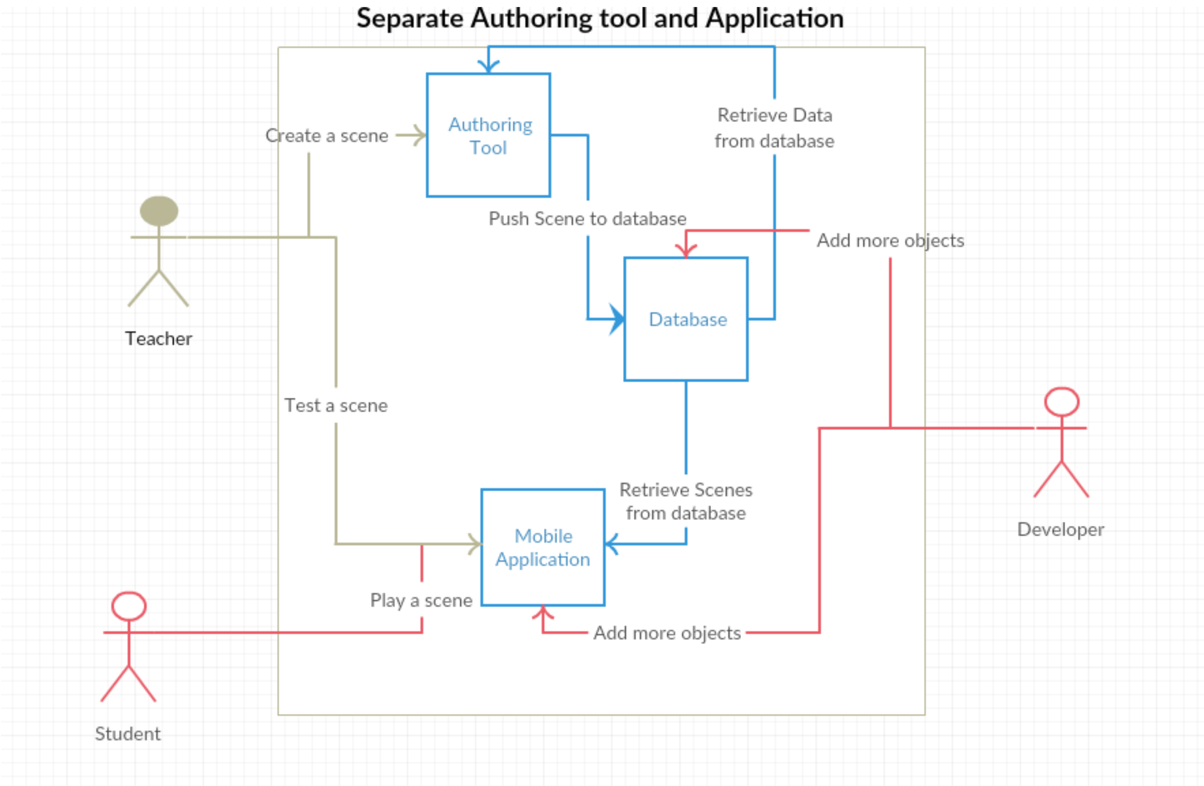


Figure 20 System Diagram for separate applications

4.2 Authoring Tool Design

The authoring tool is the application which the user will use to design a VR environment. The authoring tool is made of many parts; the most important parts can be defined into a list of:

- The Canvas or Centreboard.
- The ability to add objects.
- The ability to add tasks.
- The ability to add sounds.

4.2.1 Canvas

There are multiple different options for how the centreboard or canvas can be designed based on the multiple games of inspiration used. In this section they will be discussed and compared.

4.2.1.1 Chessboard design

The chessboard design comes from Super Mario Maker (3.1.5.4) and can be seen in Figure 8; the design gives the user a number of squares that they can fill; however they want with objects. Below would be an example of how it would look

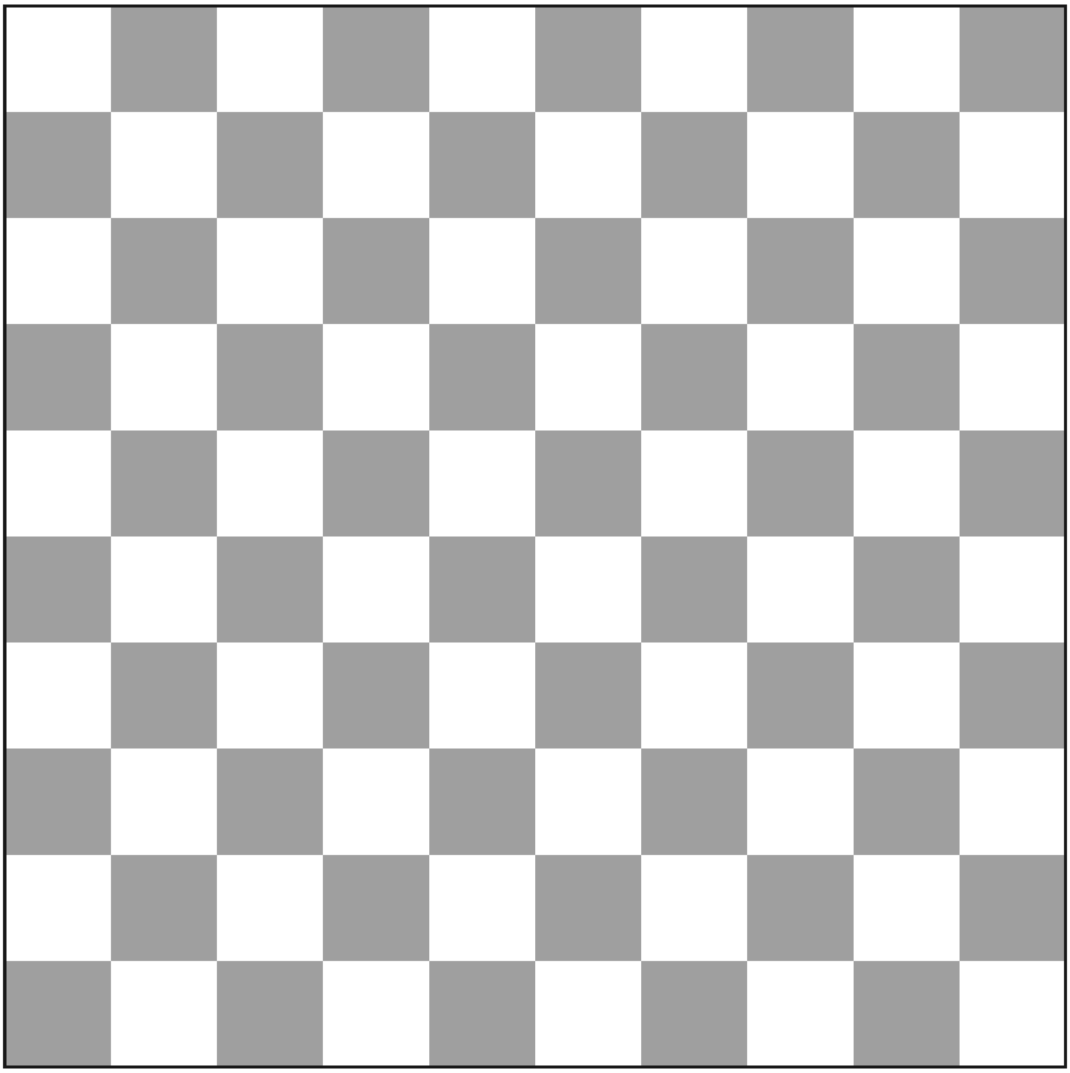


Figure 21 Chessboard Centreboard design

Advantages of such an implementation:

- Simple for the user.
- Mistakes are impossible as only available positions are allowed.
- Like what the users, would have used before.

Disadvantages:

- Lack of preciseness of object placement.
- Objects cannot be placed on top of each other.
- Hard for the user to visualise the design for the VR application.
- Objects are considered the same size even if that is not the case.

4.2.1.2 Empty canvas design

The next design comes from Line Rider (3.1.5.5) and can be seen in Figure 9, this is where the user is given an empty canvas to work with, with no guiding lines and they can place objects as they please.

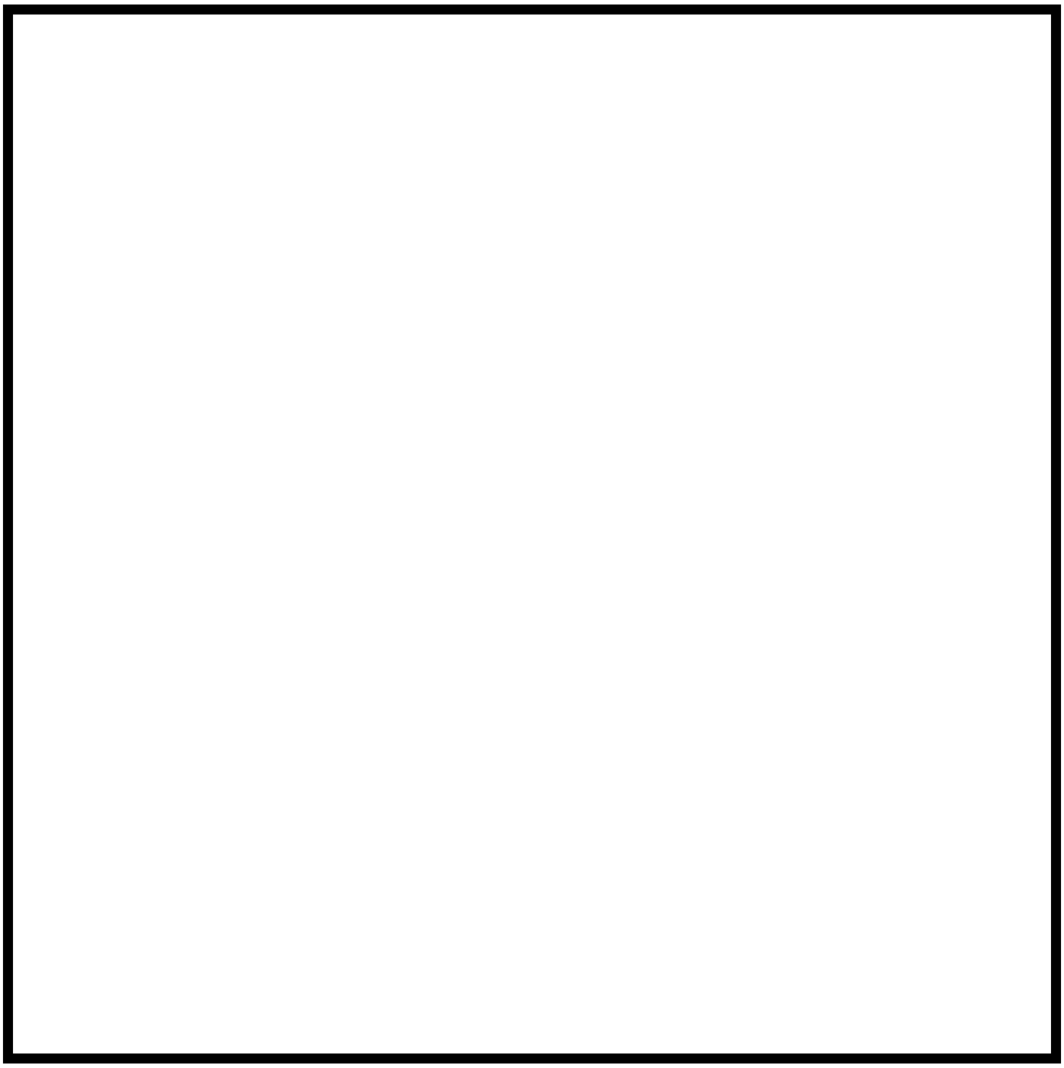


Figure 22 Empty canvas design

The advantages of making the canvas like this are:

- More preciseness in positioning then Chessboard.
- Objects can be of any size.
- It is simple for the user to interact with the canvas like this.

Disadvantages are:

- Objects cannot be placed on top of each other
- A lot of error handling is required as there is more freedom to place objects
- Hard for the user to visualise in VR

4.2.1.3 3D canvas design

The final option for the canvas design is to have it be in 3D as opposed to the other 2D options; this makes it like the G.E.C.K (3.1.5.2) and Halo Forge (3.1.5.3) example of which can be seen at Figure 6 and Figure 7. How it would look would be like this.

Figure 23 3D Canvas design

The advantages of this:

- The most precise ability to place objects.
- Easy for the user to visualise the final product.
- Can place objects on top of each other.
- Rotation is easy to do.
- Objects can be accurately sized.

The disadvantages

- More sophisticated controls for the user as they need to be able to control the camera to use the tool accurately.
- More complex implementation.
- User will not be familiar with such a tool.
- Harder to run on traditional computers as the rendering of 3D environments is required.
- Optimising of the software would be essential to be useable for the end user.

4.2.2 Adding Objects

There is only one option that came up for how to insert objects into the canvas, and this was using a Drag and Drop feature. This is where the user selects an object from a menu and drags it to the position; they want it. Once the object is on the canvas, further manipulation can be done to it.

4.2.3 Adding Tasks

The ability to add tasks for the user to complete is essential for the Scenes to have a route and flow to them, as the user requires direction to their actions within the Scene. There are two options on how to implement them into the authoring tool.

4.2.3.1 Task list
This design would be the use of an ordered list; within it, each task is listed in order. The order would be changeable, with the ability to delete tasks. The flow of the room will be doing each task one by one.

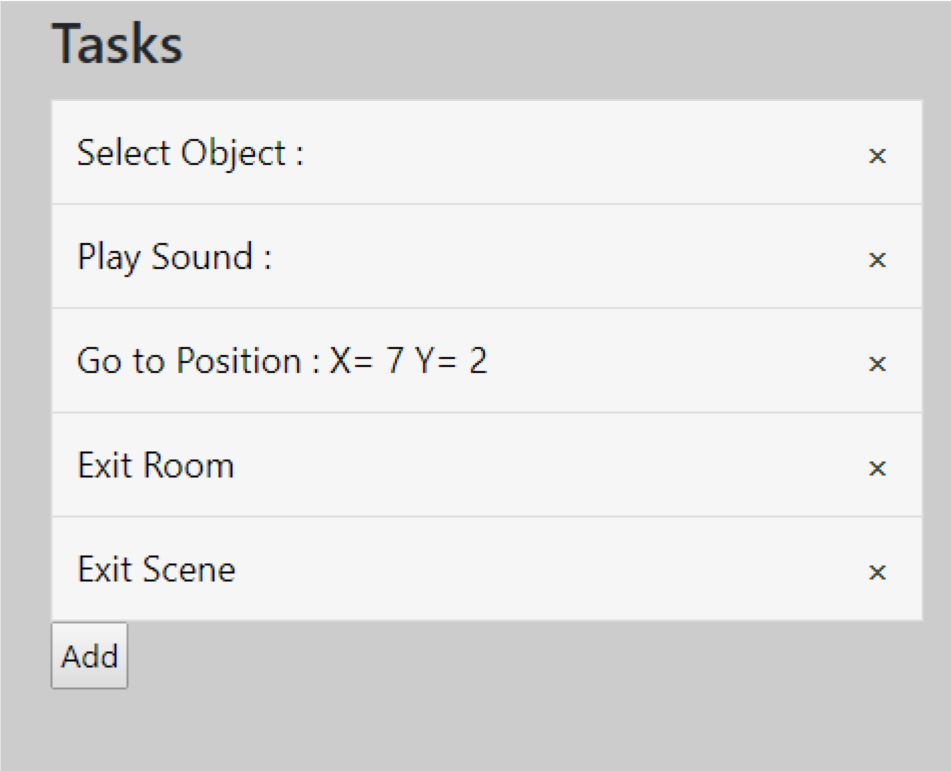


Figure 24 Task List Design

Advantages:

- Simple for the user.
- Easy to implement.
- Storage of data would be simple.

Disadvantages:

- Diverging paths and choices are impossible

4.2.3.2 Flow Chart
The alternative design option is to have a flow chart as a design tool for the user of the authoring tool to plot out how the player of the scene will move throughout the Scene. Traditional flow chart elements will be used within the tool most importantly allow for choices which can diverge the Scene into different directions. Below is an example of its implementation

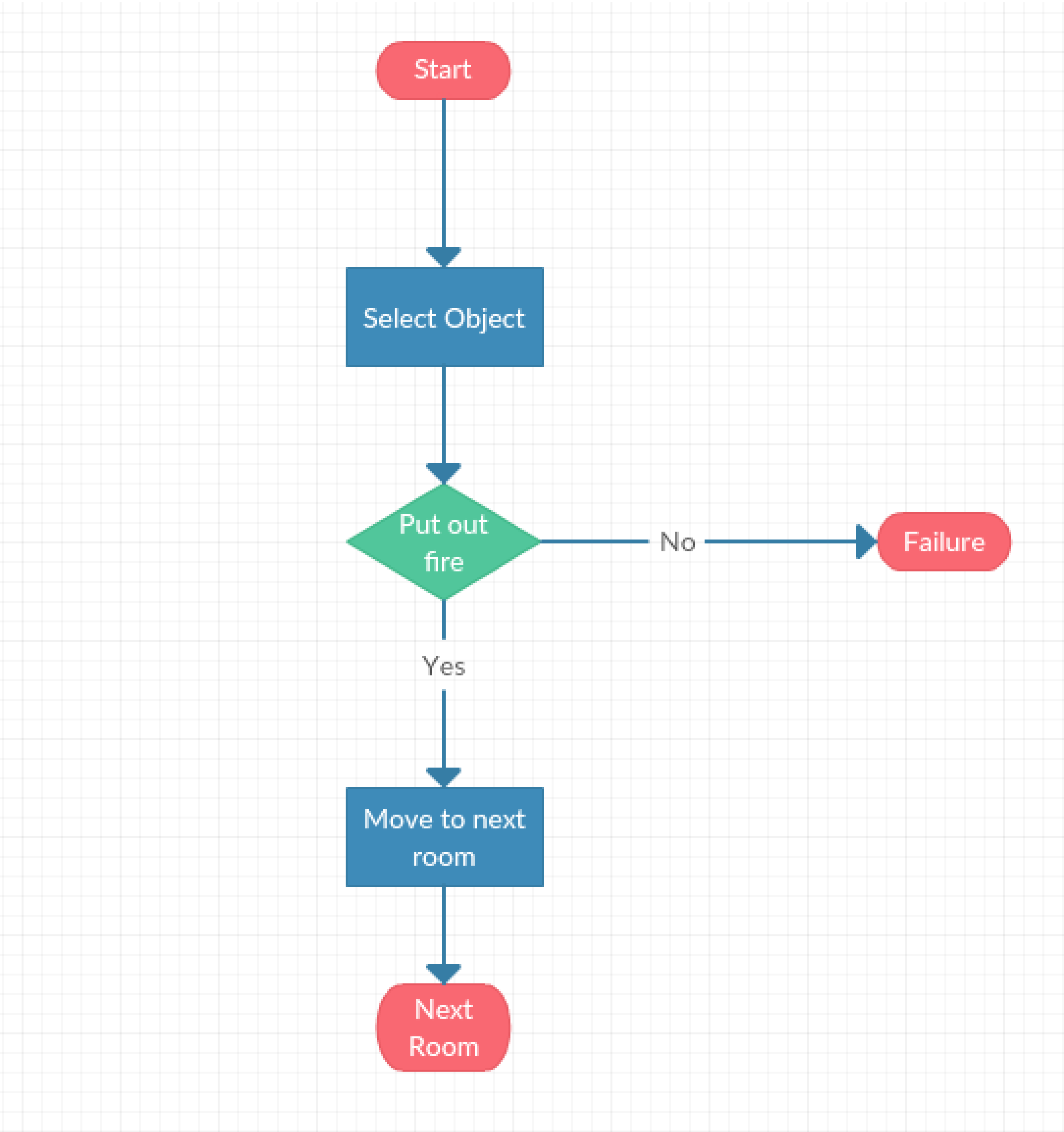


Figure 25 Flow Chart Design

Advantages:

- Allows for multiple different paths
- More flexible for users

Disadvantages

- Harder for users and can be confusing.
- Hard to store the data in a database.
- It is challenging to implement the tool.

4.2.4 Adding Sounds

There are two options for how to add sounds to the authoring tool sounds, as talked about in section **Error! Reference source not found.**, is one of the most critical elements to help with user immersion. Thus, adding it to the authoring tool is essential.

The first option is to have sound placed onto the canvas. The canvas primary is used for the user to place objects into the environment; the sounds are very similar to objects as they both have position and are in the environment. Thus, it is possible for sound to be placed on the same canvas as objects.

The second option is to have sound separate either in a different canvas or in a list. This would allow for a cleaner interface but may make it harder for users to place sound in the tool accurately.

4.2.5 UI design

The UI needs to be user-friendly so that it is usable by the users of the tool. There are multiple proposed designs for how to have the tool look like.

The first is like how Super Mario Maker does its interface with panes at the top, right and left with the canvas in the middle.

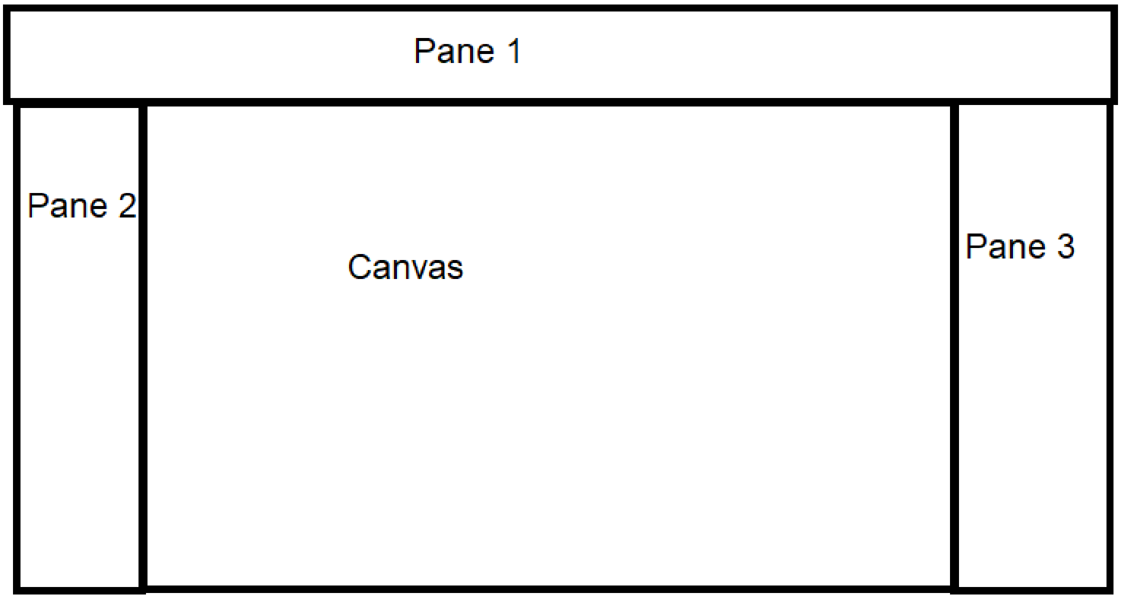


Figure 26 First UI Design

This design allows room for placement objects to go in the right pane, more options to go in the left pane and even more options at the top. The interface would work by dragging and dropping elements into the canvas or changing the settings in Pane 1 or 2.

The next option is only to have one Pane to the right of the canvas; this allows for a vast working area.

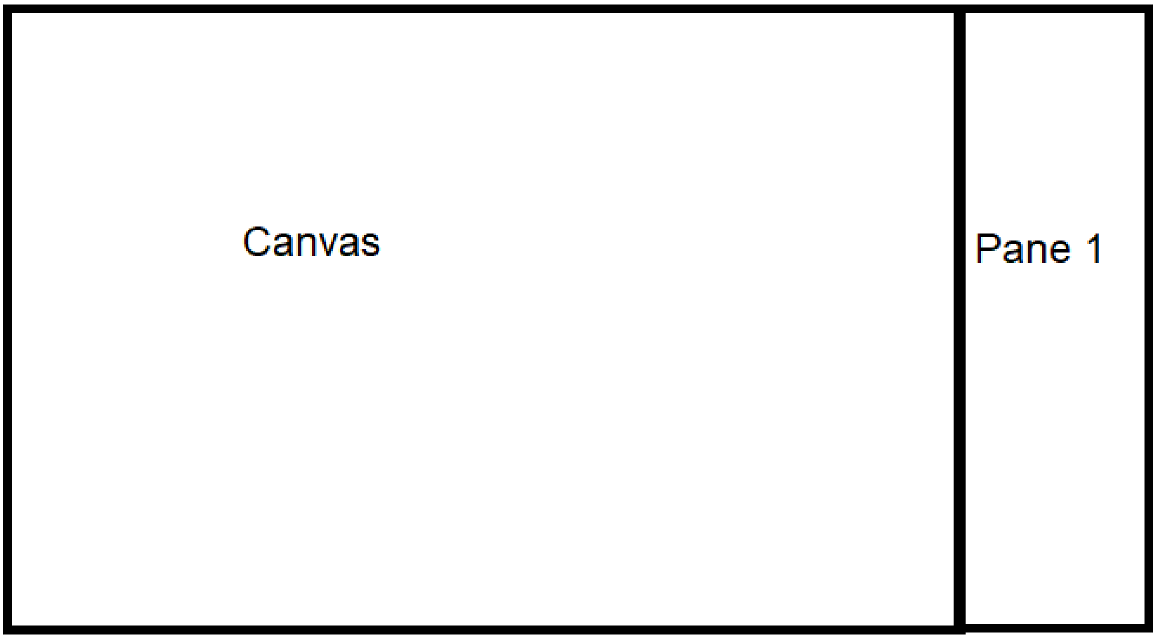


Figure 27 Second UI Design

This design would work with popup menus; this means when the user selects a part of the canvas a small menu will pop given them options this can be seen in Figure 28. The pane on the right will be used for options and settings.

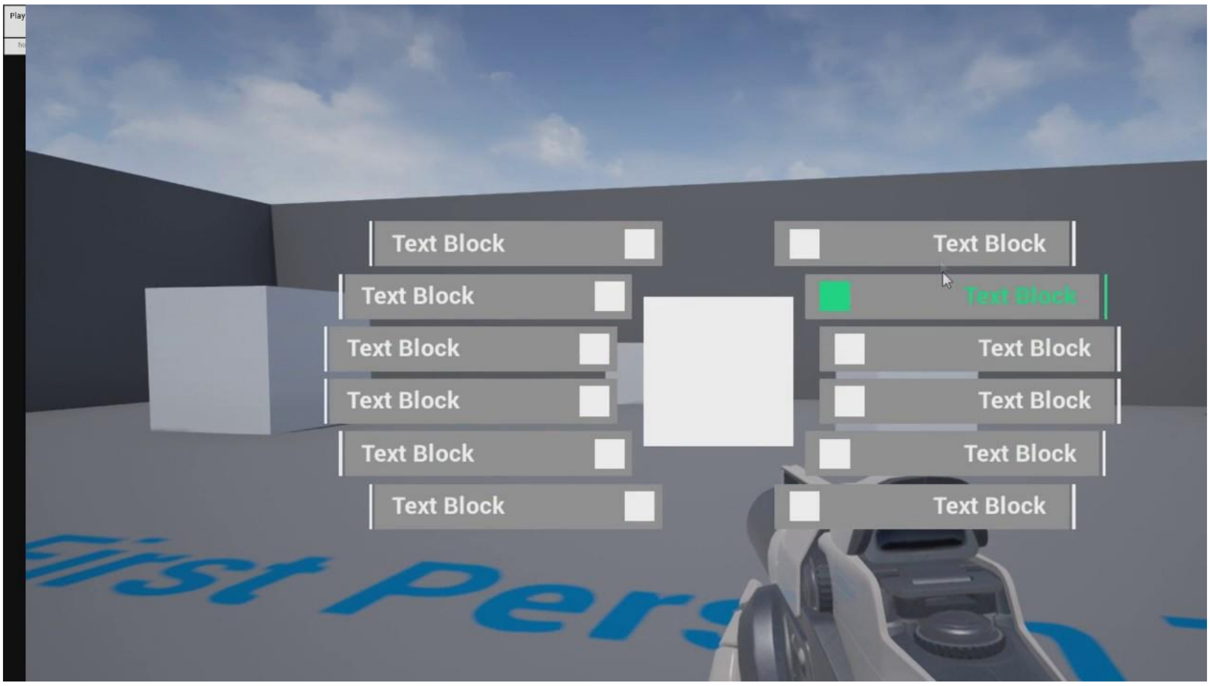


Figure 28 Popup Menu

The final proposed design takes inspiration from the G.E.C.K as seen in Figure 6. This is where the user can select and open menus and windows as they choose and place where they want, very similar to how a program like photoshop would work. The menus and windows will have settings, objects, tasks and more. Giving the user free will to have the interface look as they please.

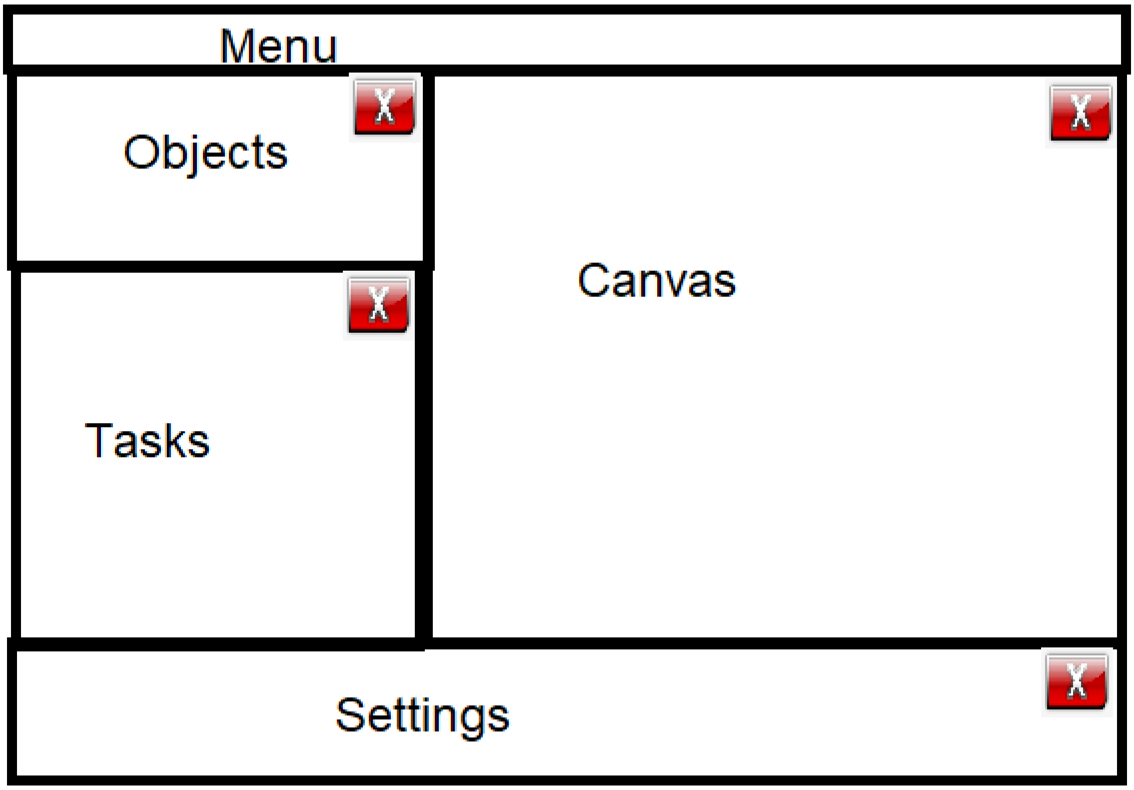


Figure 29 Third UI Design

4.2.6 Authoring Tool design decisions

The decisions on what options to go for in most cases is based on ease of implementation and user experience.

It is decided to have the authoring tool and application be separate; the main reason is that creating rooms in VR would be too difficult for the users that the project is aimed at. Furthermore, the implementation of such would be a lot more complicated than the alternative.

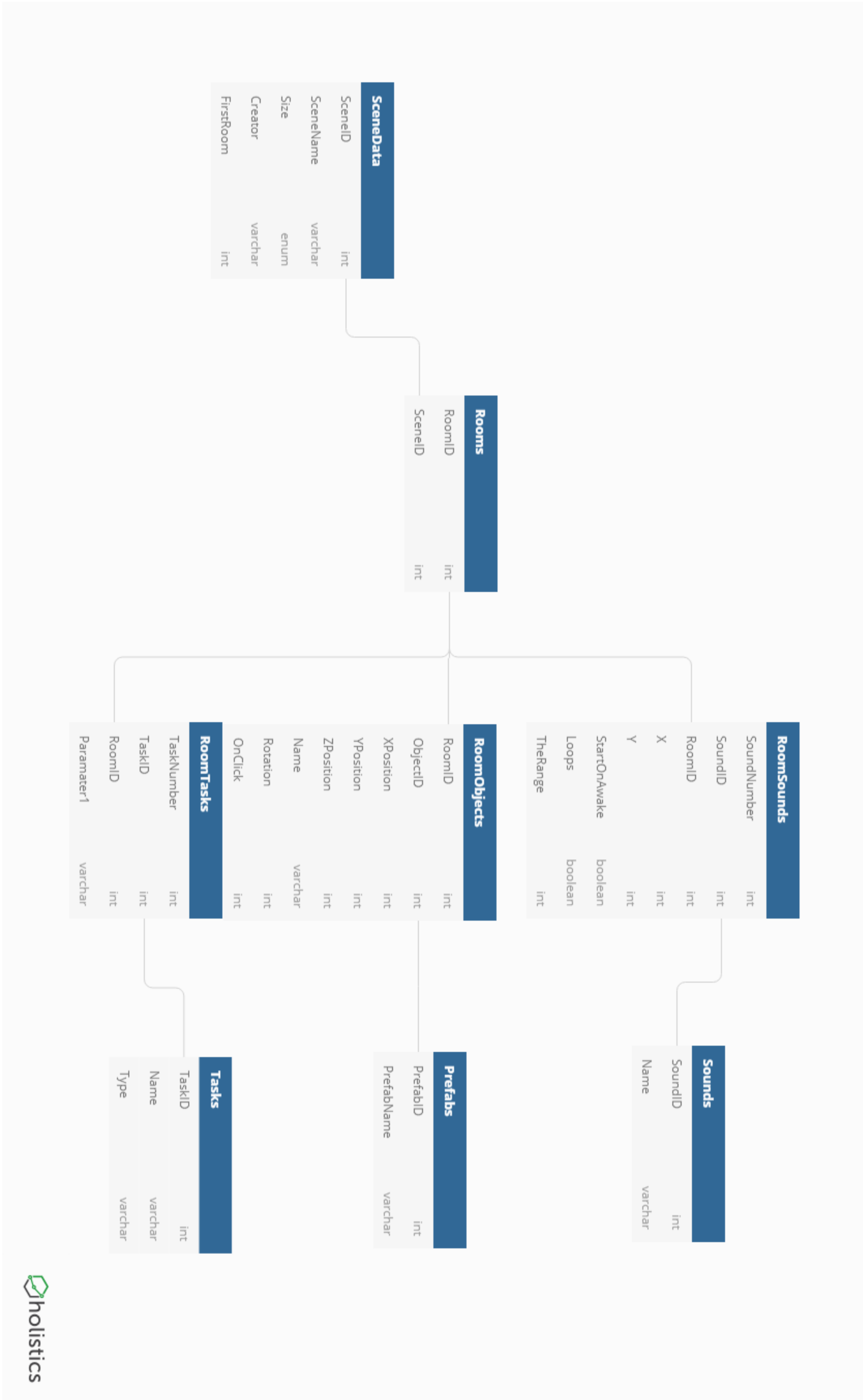
The decision on the Canvas is that having a chessboard design even though it would be less accurate then the alternatives would be the best to use as the user will find it easier to pick up and use, and the implementation of having inbuilt error checking makes it able to be developed a lot quicker.

The decision on how to implement tasks was like previous choices; A task list is a lot simpler and more comfortable to develop then the alternative even though it would not be as flexible or powerful as the alternative.

The best option for UI was decided to be using the first design as it would be the easiest for the target demographic of users and allowed for all the features to be easily implemented.

4.3 Database Design

Good database design is essential as it will be used in every part of the project and it needs to be of an efficient design to make sure time is not wasted with it. The idea is to have layers to the database where tables will only interact with the table above or below them. The first layer is to do with the Scene. This has only one table in it which holds all the overarching data of a scene. Below this layer is the room layer which only has one table which holds all the information about the rooms in a scene. This layer is needed as a scene can have multiple rooms attached to it. The next layer holds all the things that can be in a scene. In this case, that is sounds, objects and tasks. With the way, it set out its very easy to move by just adding new tables to this layer. The last layer holds all the assets that can be in the authoring tool this like the previous can be increased or added to as needed.


SceneData
SceneID int
SceneName varchar
Size enum
Creator varchar
FirstRoom int
Rooms
RoomID int
SceneID int
RoomSounds
SoundNumber int
SoundID int
RoomID int
X int
Y int
StartOnAwake boolean
Loops boolean
TheRange int
RoomObjects
RoomID int
ObjectID int
XPosition int
YPosition int
ZPosition int
Name varchar
Rotation int
OnClick int
RoomTasks
TaskNumber int
TaskID int
RoomID int
Paramater1 varchar
Sounds
SoundID int
Name varchar
Prefabs
PrefabID int
PrefabName varchar
Tasks
TaskID int
Name varchar
Type varchar
holistics

4.4 APPLICATION DESIGN

The VR application is going to be split into two sections. The first is the menu and the second is the VR environment.

4.4.1 Menu

The menu is how the user of the application will be able to choose which Environment they want to play on. The design is based on Beat Saber. The game uses an array of floating panels which contain all the information the user needs. The panels change as the user goes through the application and its menus. The user will not be able to move around when in this menu, and all the menus will be placed in the frontal direction. The user will have the option to search through all the available environments based on menu search vectors. How this would look like can be seen in Figure 10

4.4.2 VR environment

Due to the project, it is impossible to design the VR environment since that is the job of the user. What can be designed is how to interpret what the user has made on the authoring tool into VR. The choice about how to store the data is what needs to be decided. The options are between storing the data of the environments locally by downloading from the database. The other option is to pull from the database as the user is going through the application. The option that will be used is pulling from the database, although downloading has advantages of allowing the user to use the application while not connected to the internet. Pulling has much better advantages of being simpler to implement as no data storage system will need to be implemented on the VR application. The second advantage is that if the user decides to change the environment on the authoring tool. The changes will be seamlessly added if the user is pulling from the database.

4.4.3 Movement within the application

As talked about in the Movement in VR section, there are many options in what sort of movement can be incorporated into the application. Since the device is going to be geared towards mobile devices that eliminates a lot of the options that are possible as Mobiles only have a single button that can be used. The most acceptable option is to use target-based travel, this is where the user will look at a location and then when they press the action button they will be teleported there.

5 IMPLEMENTATION

5.1 SCOPE

The scope of the project is an explanation of what is required by the project, then what was added. Going through also any challenges that were encountered during implementation and lastly going through features that were not required but were implemented.

5.1.1 Features that must be added

Looking at all the requirements that were stated in section 3.5, in this section each requirement will be described and explained how it was implemented.

5.1.1.1 Adding Objects

Adding objects was one of the first objectives to complete when implementing since it was the first it came with much extra work spent setting up the authoring tool to work at all. Each part of the deliverable needs to work together to allow the user to add objects to the environment. This was done in a couple of steps

Step 1: The user needed a way of placing objects into the authoring tool, this was done through a drag and drop system, where the user will drag and drop objects into the canvas relating to where they would like them in the room.

Step 2: The objects in the room needed to be translated into a format in which they can be stored into a database.

Step 3: The object data is pushed to the database using PHP scripts and phpMyAdmin

Step 4: The user, now using the VR application, needs a way of seeing all the possible scenes they can choose from, this requires to pull all the scenes from the database from there the user can select the one they want.

Step 5: The database is called to retrieve the room data on the room that the user selected. Then that data needs to be translated so that the authoring tool can be replicated

This was the first requirement that was implemented, so there were a few challenges faced. The first challenge was incorporating the drag and drop system into the website; that process is talked about in section 5.4.1.2. Other challenges were with how to convert the data into objects in the VR application, the solution was using Unity Prefabs which are objects which are currently not in the room, but a copy of can be made and placed into the room. The last challenge was how to make sure the objects were placed in the right position; this was done by each room having an object called "0/0" this object was placed in the position that 0/0 is on the authoring tool from there all the objects that are created are placed about this object.

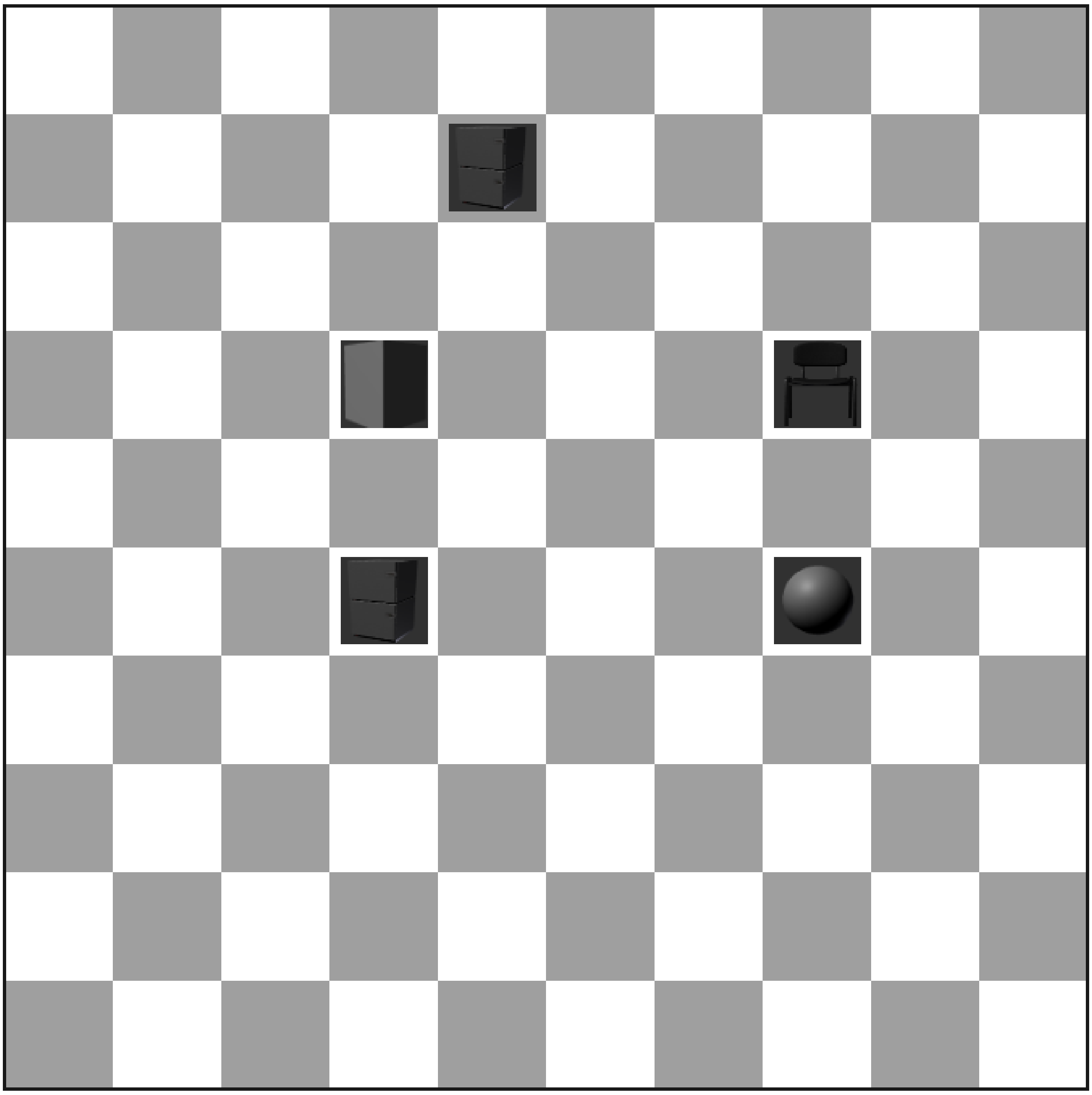


Figure 30 Adding Objects Example

5.1.1.2 Simple asset adding

The way to make adding new assets to be as simple as possible, was by reducing the amount of hardcoded data there is across all the deliverables. By doing this it would not take a long time for new assets to be added, unfortunately adding these asserts required access to the back end and no system was created to make the process possible for users this is a future work goal as stated in section 7.3.3. The current process to add a new asset is:

Step 1: Make the asset and create a prefab of it in Unity then add it to the relevant data array in the creation scene.

Step 2: Add the new asset to the relevant table in the database.

Step 3: if it is an object add a photo of it to the server so that it can be used with drag and dropping.

5.1.1.3 Adding Sounds

The process with how sounds are added to the VR application from the authoring is like how it is done for objects expect with the way that the user adds sound to the authoring tool.

It was decided that sounds should be separate from objects on the canvas, so a new screen was made which will pop out at the user which will allow them to add sounds in it. Within the pop out the user can select the type of sound, the location of the sound, whether the sound will start when the room starts and whether the sound loops. Once chosen they would be added to a list when the scene is saved to the database the contents of the list is also saved.

One challenge was faced during this part, and it was getting the user to be able to hear the sounds, this was fixed by changing the audio settings to do with the user in the VR application.

5.1.1.4 Adding Tasks

Adding tasks was one of the more challenging implementation objectives as it changed a lot about the initial development of the tool. Section 5.4.1.3 goes more in-depth in how tasks were created in the authoring tool. After the tasks were adding to the database challenges was faced in how to translate them within the application. An issue came up with specific tasks not functioning, these being the tasks that play sound, the reason for this was because the program was trying to assign a script to an object that had not been created yet. The solution of which was to make the creation of tasks to be the final thing that the Application does when creating rooms.

The next challenge was how to differentiate sounds from each other if they have are the same, for example, if there are two fire alarm sounds in a room and the user wants one to go off at the start and the second to go off after the user has moved to a location. As the application as there was no way to select the second sound, a solution was coming up where when a sound is added to the authoring tool, it is given a random id and by using those is how sound is referenced in the application.

5.1.1.5 Movement

Movement is an important part, and the addition of it into the program was simple, a Unity store asset called VR Teleporter, allowed for the inclusion of a teleporting movement without any new code to be written.

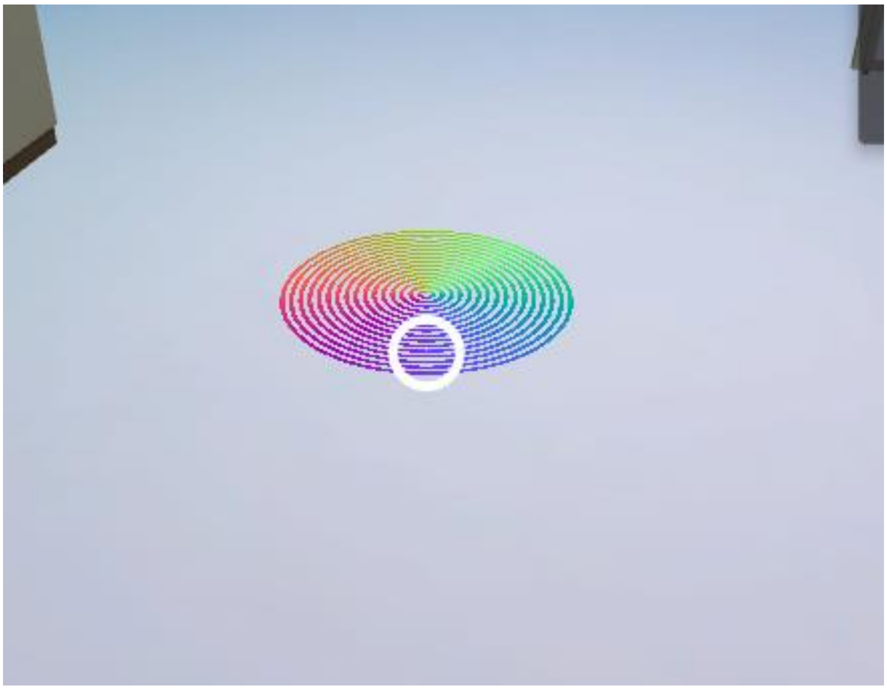


Figure 31 VR Teleporter example

5.1.2 Features that were added that were not required

During development, some features that initially were not planned on being included were included due to changes in what the health and safety wanted and improving the overall quality of the deliverable.

5.1.2.1 Multiple Rooms

The feature of allowing the user the ability to add multiple rooms was an early inclusion to so the user can have more space to create their environment by spreading across multiple rooms. Section 5.4.1.1 goes into more depth on the implementation of the feature.

The challenges faced was changing the way data stored in the project, when there was only one room data only needed to be stored on the canvas. When adding additional rooms data is needed to be stored in other locations, in this case, using cookies. This change also made saving data to the database more difficult as there was more data to deal with at different levels of a data structure, section 5.3 goes more in-depth on how the data structure works and looks.

The next challenge was how to incorporate multiple rooms into the VR application; the way rooms were made previously is when the Unity Scene was loaded the room id was passed, and a get request was made to the database to get all data related to that room id. To allow for more rooms, when the user exits the room the original room id is incremented then the Scene is reloaded, and now different data is returned from the database. This method of room ID is changed later in section 5.1.2.3

5.1.2.2 Retrieve Previous Rooms

The feature was added as a way of confirming if what had been created in the authoring tool was actually saved to the database but later on especially when combined with the Update Rooms feature talked about in section 5.1.2.3 gave more functionality to the user as they can save the progress of their work.

The way the user can retrieve previous room data is a complete reverse to how data is saved to the database. It is done by rebuilding the room with the same data that was saved to the room.

5.1.2.3 Update Rooms

The feature of letting the user create a room, then come back and make changes to it is a massive increase to the user experience as the user can fine-tune their Scene by making small changes and improving it.

The implementation of how to update the scene is by deleting all the data of the original scene and then rebuilding the scene entirely with the changes that the user has made.

A challenge came with how the original incarnation of multiple rooms, the issue came when updating previous rooms; the user may add a new room when there was not one before, this new room would overlap with the other completely different rooms which would cause errors in the VR application. The solution to this was to change how rooms are stored on the database; pointers were used to solve this. Each room has a pointer which points to the next room in the scene; this allows as many or as few rooms as the user wants without messing up the database.

5.1.2.4 Change Room Location

By adding the feature of having multiple rooms the next logical extension is the ability to change what the overall room look like, currently, the user can only change what is inside of the room they cannot change what the overall room looks like.

Adding this feature meant some more of the code had to change, the first was changing all hardcoded instances which relate to the size of the room, before the size of the room was always 10x10, now that could change based on the user choice of room.

Another challenge was when creating the room, the overall look of the room was already loaded when a user entered the room but now it had to load the same way that objects are except it must be done before so they can be placed accurately within the room.

5.1.2.5 Properties of Objects

5.1.3 Features that were not added

Unfortunately, due to choices made in the design phase or due to lack of time, some features were not able to be implemented. They will be gone through here with ideas on how to implement them later.

5.1.3.1 Choices

Implementing choices into the authoring tool and application is very difficult due to the complexity that comes from the inclusion. As the deliverable currently stands tasks are done in a linear order, 1 – 2 – 3 and so on by adding choices tasks are no longer linear they can go 1 – 4 – 2 – 8. Tasks can be missed and not happen due to choices by the user. This makes translating the data very difficult which was the reason why choices did not make it into the final deliverable.

in section 4.2.3.2 the idea of a flow chart was proposed, if there were found a way of storing and translating the task data, then the next step would be to add a flow chart system into the authoring tool.

5.1.3.2 Interaction with hands

Due to choices made about the device used in deliverable, the ability to interact with the user's hands in the environment was lost. The device being used is a mobile phone which has no controller to replicate the use of hands. Future work can be to port the application to devices which can use controllers like the Oculus GO or the HTC Vive.

5.2 COMMUNICATION OVERVIEW

This section will go in-depth in the communication paths and stratagems that are being used to work with all the applications. It is essential to show that the VR application does not directly communicate with the Authoring Tool they both go through the medium of the database which then facilitates all requests to and from each. In the applications current state, there are only three vectors of communication. The first is the authoring tool pushing data to the database; the second is the authoring tool requesting data from the database and finally is the VR application requesting data from the database. The Application does not push any data up to the database; this option could be added if the functionality is added to the application which facilitates a reason for it.

5.2.1 Authoring tool pushing data to the database

This vector path is only used to do one thing, and that is to add new scenes that the user has created to the database. Below is the step by step process of how this happens.

Step 1: The user creates their rooms with all the tasks, objects and sounds that they want.

Step 2: The current room that the user is working on is saved to the cookies that are used on the website (cookies are explained in section 5.3)

Step 3: The data from all the cookies are placed into a single data structure.

Step 4: This data structure which holds all the data of the room is passed to a PHP file with another data structure which holds overall data of the scene, this is like the name and size of the scene. This PHP file is called asynchronously by using an AJAX request.

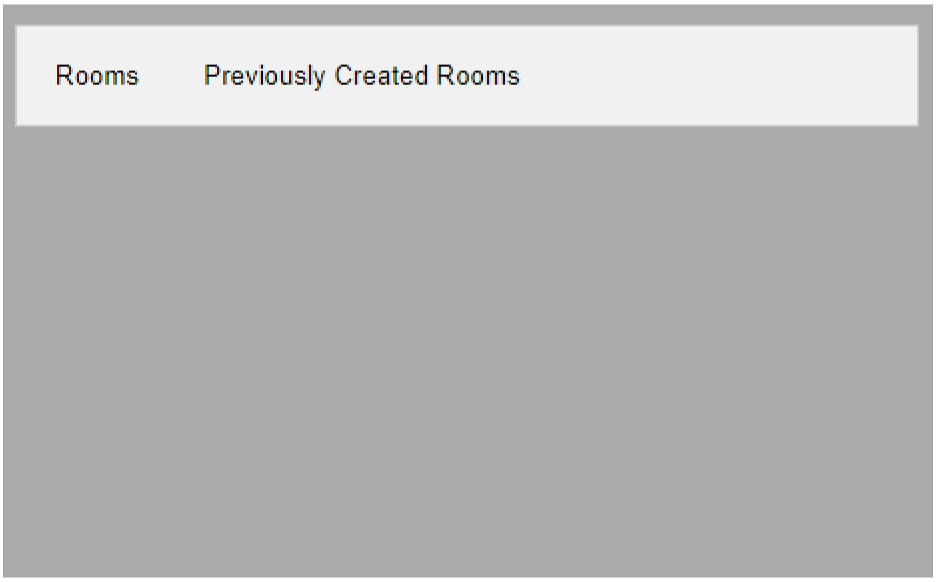
Step 5: All the data is put into correct MYSQL statements to be able to add them to the database. Care needs to be taken when dealing with the value of firstroom which is associated with every scene. As the amount of rooms already made is not known to the website so when passing this data, need to return the amount of already made room so later the database can return the right room when requesting a specific scene.

Step 6: if no errors have occurred the PHP returns that the room was successfully created.

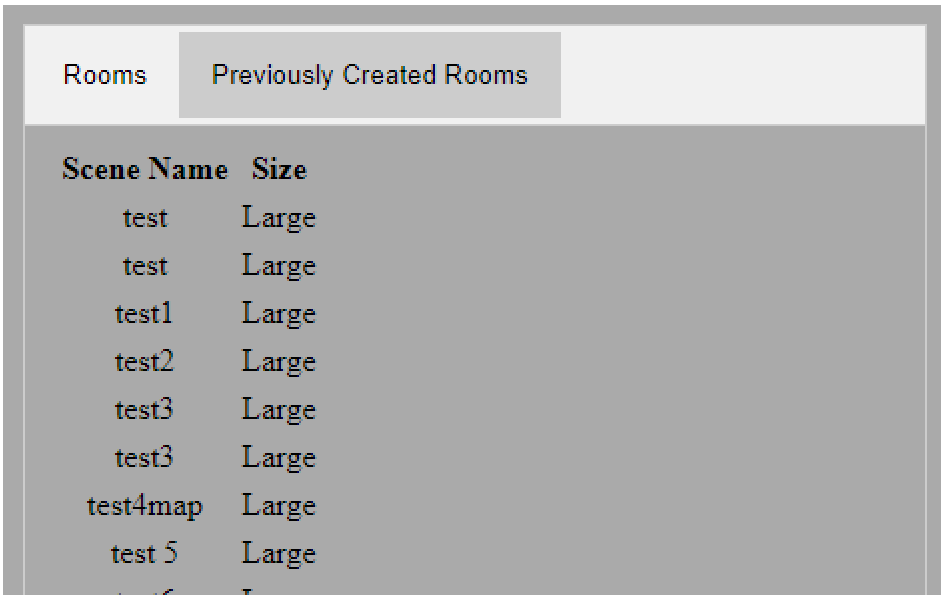
5.2.2 Authoring tool requesting data from the database

A lot of the data on the website is first fetched from the database; the reason for this is to have as little hardcoding as possible on the website. This means that changing items in the database will be reflected all over the project.

When the website is loaded, three PHP requests are made to retrieve data from the database. The data being retrieved are the list of tasks, list of objects and the list of sounds. These data sets are used to populate the relevant parts of the website.



When the user clicks on "Previously Created Rooms" another PHP request is made to retrieve all the previously created rooms into a list.



Now on the website, there is a list of all the rooms that the user has previously created. When a user clicks on any of the previous rooms, then another PHP request will be sent. This PHP request is to retrieve all the data of the room that the user clicked.

Step 1: Remove all objects from the current board

Step 2: Pass the values of the room that the user has selected to the PHP function

Step 3: Connect to the database

Step 4: Return all the objects, tasks and sounds and pass them back to the website

Step 5: Recreate all the objects, tasks and sounds into the website.

5.2.3 VR Application requesting data from the database

The VR application does two types of requests to the database. The first is the request to get the list of created scenes. The second request is to retrieve the room which the user is entering.

When the user starts up the VR application, a PHP request is sent to the database; this returns the scene data for each of the scenes that the user has created. This data is put into a table that the user can interact with. This table is a big part of the menu system. When a user selects an option from the menu, the user is moved into a new room where the environments that they made in the authoring can be generated.

When a user enters the "creation room" this is the room in which the environment is generated that the user created is made. When entered a value is passed which lines up to the correct room that the user wanted to enter. With this value, a PHP request is made. This request returns all the objects, tasks and sounds that are associated with that room. The PHP request passes all this data back to the VR application which creates all the items and places them into the environment.

5.3 DATA STRUCTURES

In this section, will be a quick overview of all the data structures which are extremely vital to how the whole project works as values in one will be associated with specific other data structures.

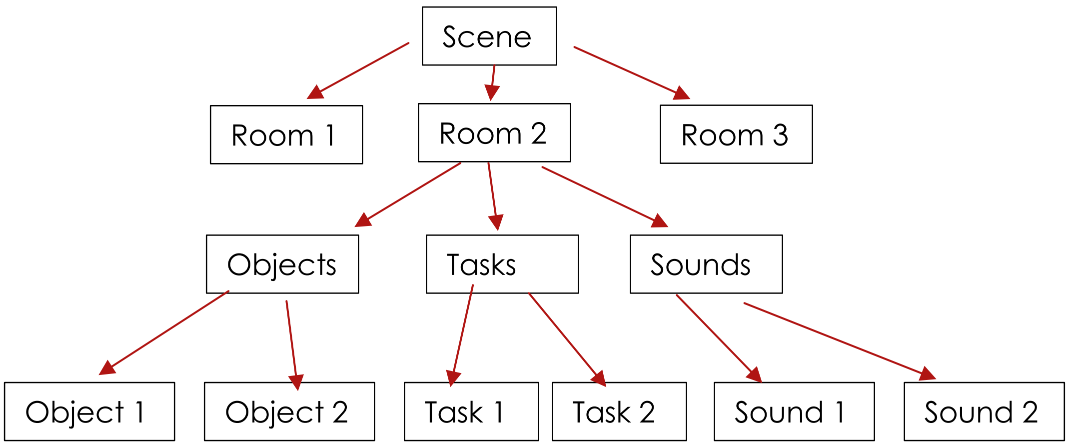


Figure 32 Array Overview

The diagram above is a representation of the array structure that is used within the authoring tool to hold all the data that the user has made. The structure is a 4-dimensional array. The best way to explain is to show what is returned as more parameters are added to the array. The array is called Scene; Scene [0] will return all the data relating to the first room in the scene. Scene[0][0] will return all the objects in the first room in the scene. Scene[0][0][0] will return an array which holds all the information about the first object in the first room in the scene. Finally, Scene[0][0][0][0] will return the object type of the first object in the first room in the scene.

All the higher arrays only hold the number of objects, rooms, tasks etc. the only array that is unique in that sense is the lowest arrays which indicate what information is held about objects, tasks and sounds. Starting with objects

Table 1 Object Array

0	1	2	3	4	5	6	7
Object Type	X	Y	Z	RoomID	Name	Rotation	Onclick

Object Type: This value is connected to the Prefab table and refers to the type of object that is wanted to be created.

X; Y; Z: These values refer to the x,y,z position of the object within the room.

RoomID: Refers to a value in the database, as every room has a unique number which differs it to all other rooms in the whole database.

Name: This is the name given to the object; the user can change this.

Rotation: The rotation on the object in the Y rotational axis.

Onclick: a numeric value which refers to what happens when an object is selected in the VR application.

The next array is the one for Tasks.

Table 2 Task Array

0	1	2
TaskID	Parameter	RoomID

TaskID: The value is connected to the task table, and it states what type of task is being chosen.

Parameter: holds the value that is associated with the task.

RoomID: Refers to the unique value for the room associated with the task.

The last array is for Sounds.

Table 3 Sound Array

0	1	2	3	4	5	6
SoundID	X	Y	StartOnAwake	Loop	Range	RoomID

SoundID: Refers to the Sounds table, refers to what type of sound has been chosen.

X; Y: X and Y position of the sound in the environment.

StartOnAwake: A Boolean on whether the sound should start as soon as the environment is loaded.

Loop: A Boolean on whether the sound should loop after playing once.

Range: How far the sound should propagate from its source.

RoomID: The room which is associated with the sound.

On the authoring tool, the way that all this data is stored is through cookies. Cookies allow for data to be stored on the website, even if the user were to leave the website. The authoring tool uses separate cookies for each room of the scene.

5.4 Code Breakdown and Highlighting

In this section, there will be some highlighting and breakdown of interesting or well-implemented functions in the project.

5.4.1 Authoring Tool

5.4.1.1 Multiple Room Implementation

Working out how to allow the user to make multiple rooms was one of the earliest coding challenges faced in the project. This was an essential inclusion into the authoring tool as the user would be forced only to have one room of which to play around with, by allowing multiple rooms the user has more freedom to make the environments they want and allows for an actual path through the scene.

The way this was implemented was with the use of a Javascript Library called Javascript Cookie (Link). This library allows for easy use of cookies on a website; cookies act as a source of storage on the website. Their role is to hold data about each room so if room changes happen the data is not lost. To be more specific the steps involved.

Step 1: The user has added objects, tasks and sounds to the main board.

Step 2: The user clicks on a different room from the one there are currently editing.

Step 3: The name of the room the user wants to enter is passed to the function changeroom(newRoomName)

Step 4: The newRoomName is compared with the currentRoom variable to see if the user has clicked on a different room.

Step 5: The current cookie for the current room is removed

Step 6: The contents of the current room are put into an array and saved to that cookie.

Step 7: The contents of the cookie which is associated with the room the user wants to go to is saved to an array.

Step 8: The board is cleared of all objects, tasks and sounds.

Step 9: The contents of the new room cookie are put onto the board.

5.4.1.2 Drag and Drop

Adding drag and drop to the centreboard was another critical feature that needed to be implemented, to allow the user to have an enjoyable experience on the website.

The way that drag and drop are added to is through a new HTML5 attribute which can be applied to elements of a website. In this case, the attribute is added to the images which represent objects in the environment. Adding the attributes is all that is required to have a semi-working drag and drop system. However, for the project some more features were wanted, to be more specific the ability to swap two objects around and the ability to remove an object by placing it over a trash can inferring this will delete the object. Implementing these took some more work.

To implement the ability to remove objects was simple when implementing the original drag and drop system there need to be a drop function within it a copy of the draggable object is copied to the dropping element. Adding the ability to remove is done by creating a new function which is called only when the drop element is the trash can.

To allow for a swap is more difficult due to there being several cases which must be accounted for in the code.

The first case is the normal one where the destination square is empty, and the object is being dragged is from the object list, when this happens, the new object in the board is given an id to be able to distinguish it from other objects.

The second case is where an object already in the centreboard is moved to another square in the board which is empty; in this case, copy the object over to the new square. A new id does not need to be made.

The third case is where an object is already in the centreboard and is moved to the same square it is in. In this case, nothing should happen.

The final case is where an object already in the centreboard is moved to a square which has another object within it, you make copies of both the original object and the destination object, you get the locations of each object, delete the original versions of the objects and then attach the copies the other way around.

5.4.1.3 Tasks panel and task creation

The task panel and especially the act of task creation are very dynamic, their look and function are based on choices made by the user.

Based on choices in the first drop down will dynamically affect what appears in the second box, furthermore what is in the second box will change what data is added about a task when it is added to the authoring tool. The changes are not unique to each task as tasks may warrant a similar interface. In the database there is a list of all the types of tasks that can be done; each entry in the list has the type of interface that is required to choose the parameters for the task. On to the interfaces

The first is called "Objects" this interface returns a drop-down list of all the objects in the current room distinguished by the name the user has given them. This interface is used for the Task "Select Object" this task is about clicking on an object in the environment the drop-down list allows you to choose which object. When it comes to adding a task of this form, it saves it to the task list in the form of "Select Object : [Name of object] as its parameter it saves the name of the object.

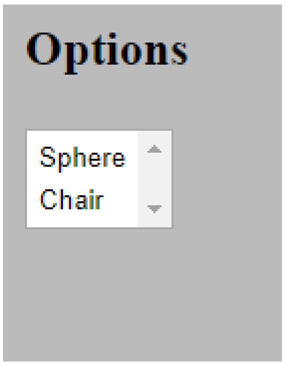


Figure 33 Objects Task Creation

The second is called "Chessboard" this interface returns a smaller version of the main centreboard that is used in the main part of the website. This interface can select positions on the environment. This is used in the task "Go to position" which asks the user to move to a certain position in the environment. It is saved in the task list in the form of "Go to position: X [x position on the map] Y: [y position on the map]. This saves the X and Y positions as the parameter.

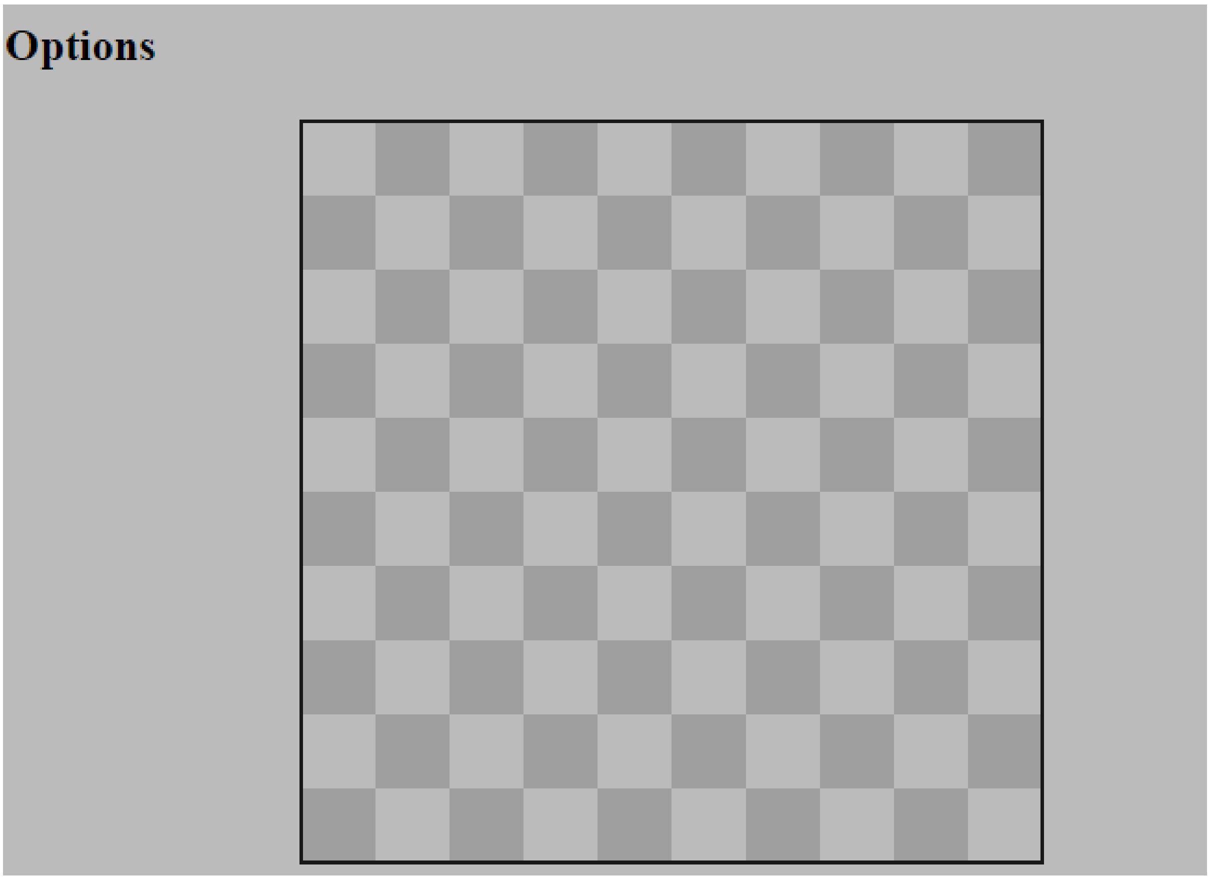


Figure 34 Chessboard Task Creation

The third is called "Nothing" this interface returns a space as no interface is required. This is used for the "Exit room" task as no parameters are required to be chosen.

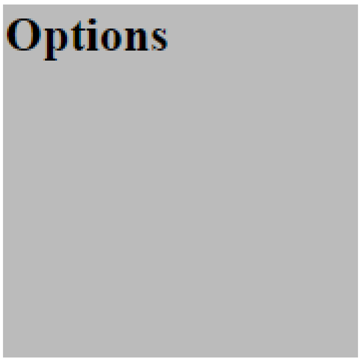


Figure 35 Nothing Task Creation

The last is called "Sound" this interface returns all the sounds that are in the canvas at the moment, then allows the user to select the one they intend the task to be about.

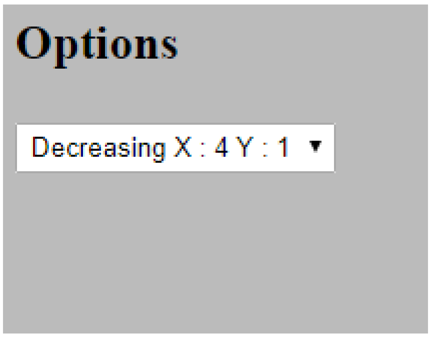


Figure 36 Sound Task Creation

6 EVALUATION

In this section of the report, we will be evaluating the implementation of the requirements. We will be testing the implementation with some users to see if the requirement has been successful, if not it will give ideas on where the project can be possibly improved.

6.1 REQUIREMENTS

Here we will have a table which will indicate all the requirements that must be done for the project with ways that they can be tested with expected outcomes.

Test Number	Requirement	Test Plan	Expected Outcome
1	To save the user's creation into the database.	Create extreme and regular rooms and comparing them with what is saved in the database.	The authoring tool will be able to handle any amount of data entered
2	Objects placed in the authoring tool match their location in the VR environment	Place objects on the authoring tool and compare them with where they appear in the VR	The VR and authoring tool will work together seamlessly for accurate replication
3	That all the tasks work as intended	Test each task with extreme and normal variables to see how they cope	All tasks should be able to deal with normal data, but extreme data may cause issues
4	The order of the tasks is kept	Testing sets of tasks in different orders and combinations to see if the order is always kept	For normal situations the VR application should be able to handle any order of tasks, Issue may occur when duplicate or incompletable tasks are added to the list.

6.2 TEST PLAN

The testing of the project is going to be requirements testing that will be self-evaluated based on the requirements stated in the previous section.

6.2.1 Requirements Testing

The goal of requirement testing is to see if the project was successful in achieving what it set out to achieve. In the previous section is a table with all the requirements that will be tested with how they will be tested and expected outcomes. The testing will be conducted internally, all the values that were used and the results of each requirement test will be recorded and placed into tables and graphs.

6.3 TEST RESULTS

In this section will be all the test results gained from the testing

6.3.1 Requirements Test Results

6.3.1.1 Test 1

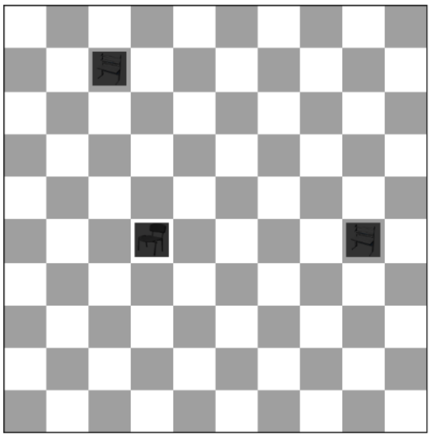
Goal: To save the user's creation into the database.

Test 1: Normal data added to the authoring tool

Test 2: No data added to the authoring tool

Test 3: Maximum data added to the authoring tool

6.3.1.1.1 Test 1 Results



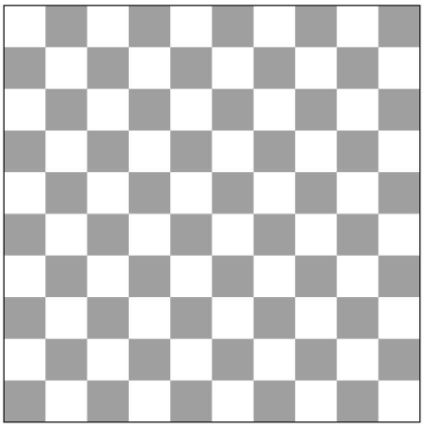
+ Options

	RoomID	ObjectID	XPosition	YPosition	ZPosition	Name	Rotation	Onclick
Edit Copy Delete	222	1	2	0	1	Table	0	0
Edit Copy Delete	222	0	3	0	5	Chair	0	0
Edit Copy Delete	222	1	8	0	5	Table	0	0

Check All With selected: Edit Delete Export

The database assumes the most top left the square to be (0,0), looking at the data the first table object is given the X position of two which means the three squares across and the Z position of one which means two squares down. So, the data is correctly replicated.

6.3.1.1.2 Test 2 Results



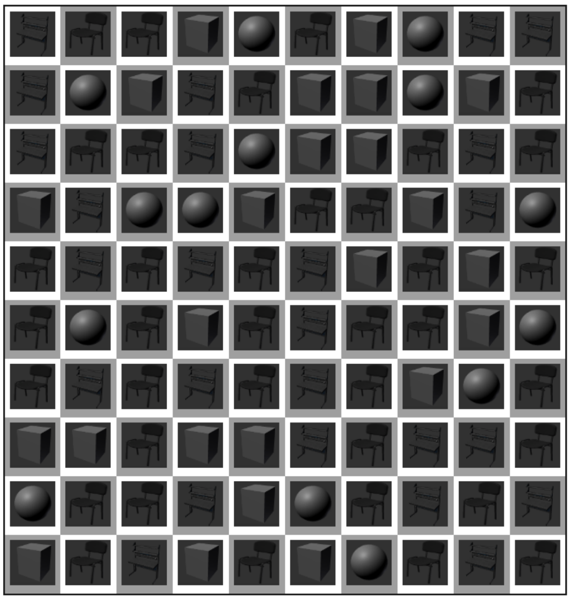
+ Options

	SceneID	SceneName	Size	Creator	FirstRoom
Edit Copy Delete	320	Test 2	Large	Kareem	0

	RoomID	SceneID
Edit Copy Delete	156	271

When no objects are added to the environment, the scene is still made and is seen in the database. However, a weird occurrence happens where it says the first room that is associated with that scene is Room 0; Room 0 does not exist in the database so, this occurrence causes an issue. If a room has no objects it ceases to exist in the database, this is not a common issue as its expected most users always to have objects in the room. One occurrence possible is if a room has no objects but instead has just sound based on this the room will be inaccessible. Thankfully due to how the database is set up Room 0 cannot be filled with another room which would cause a lot of issues.

6.3.1.1.3 Test 3 results



	SceneID	SceneName	Size	Creator	FirstRoom
Edit Copy Delete	321	Test 3	Large	Kareem	0

Once again like what happens with having no data, having maximum data appears to confuse the authoring tool and stop it from working. As it stops it from creating a room and no objects are added to the database. The results of these two last tests will be investigated in the bug fixing process.

6.3.1.2 Test 2

Goal: Objects placed in the authoring tool match their location in the VR environment

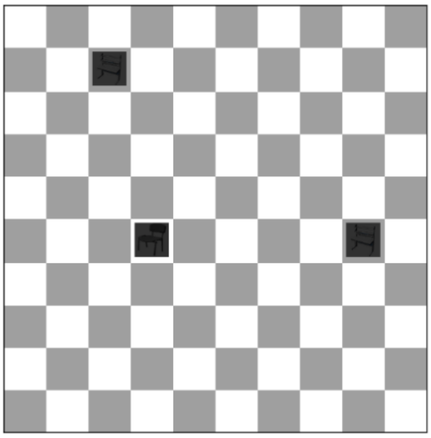
Test 1: Normal data added to the authoring tool

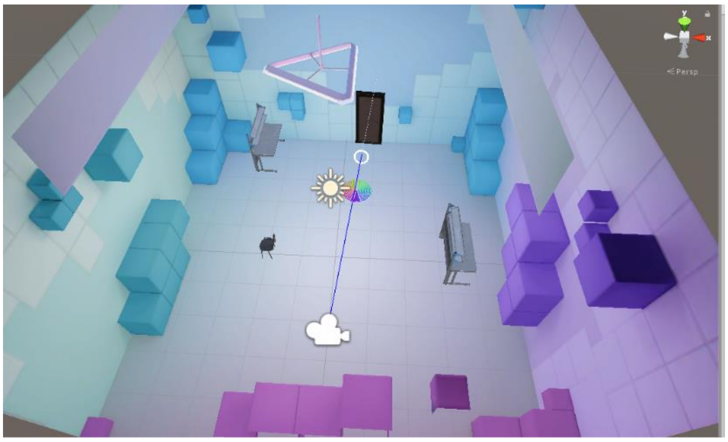
Test 2: No data added to the authoring tool

Test 3: Maximum data added to the authoring tool

The data used in Test 1 will be used in this test to see if the full use of the whole project is possible.

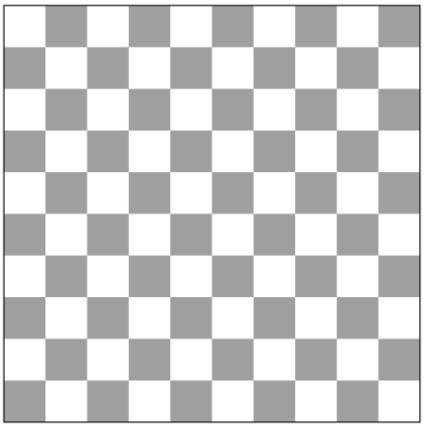
6.3.1.2.1 Test 1

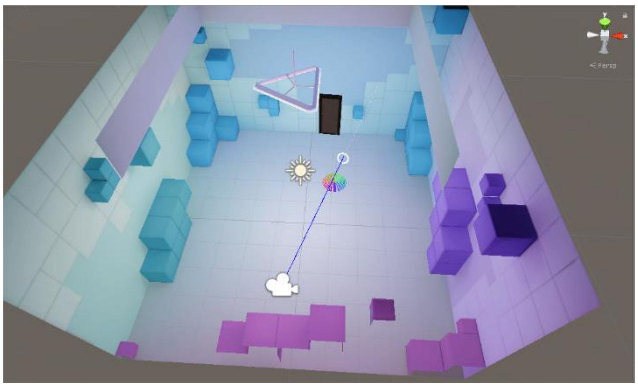




The replication of the authoring tool looks very accurate on the VR application.

6.3.1.2.2 Test 2

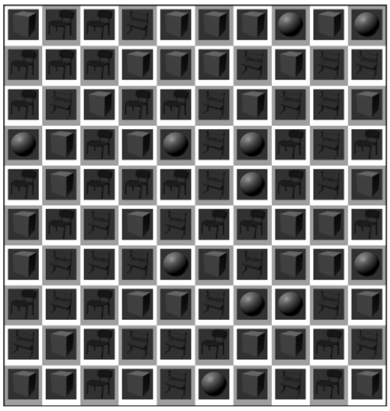
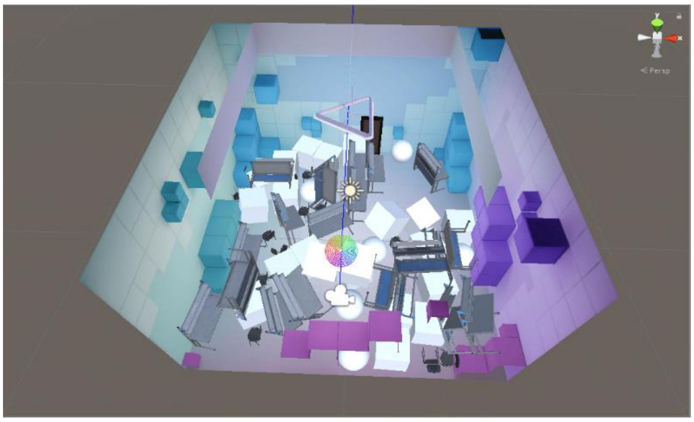




There is nothing on the authoring tool, and there is nothing on the right, so that is a success.

6.3.1.2.3 Test 3

In the previous test 3 where a max room bugged out, in response to that the max test was remade to see how much can be added before it bugged out, but this time it allowed a full room with no issue which means this issue will be hard to debug as it isn't consistent.

The results as seen, are a failure. It is hard to define as putting all those objects into one room caused a big explosion of a mess as all the objects came to be one after each other banging and rebounding off themselves. The issue is that each object is not the size of the square, many are smaller, and many are larger; this causes overlaps which make collisions happen. An improvement would be to either force all objects to be the size of a square or less or make objects more reflect their actual size in the authoring tool.

6.3.1.3 Test 3

Goal: That all the tasks work as intended

Test 1: Select Object

Test 2: Go to Position

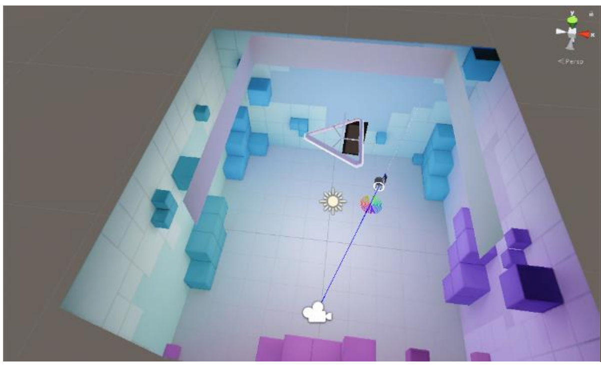
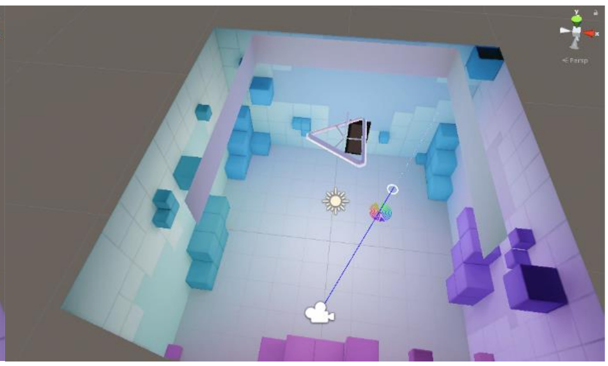
Test 3: Exit Room

Test 4: Play Sound

Test 5: Stop Sound

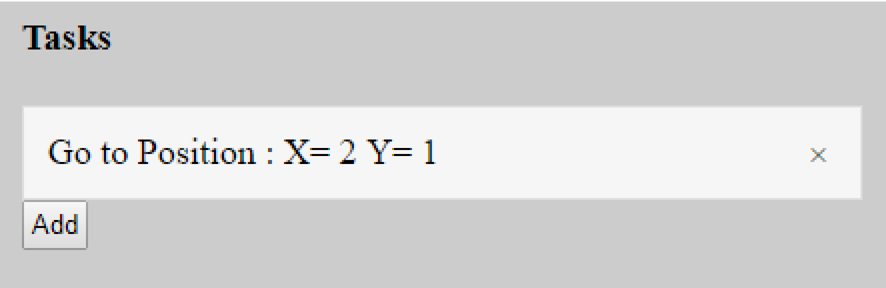
6.3.1.3.1 Test 1

Before interacting with the chair. After interacting with the chair

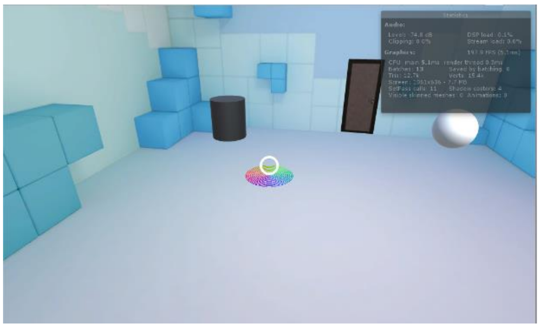
 

Successful test as the Select-Object task works as intended

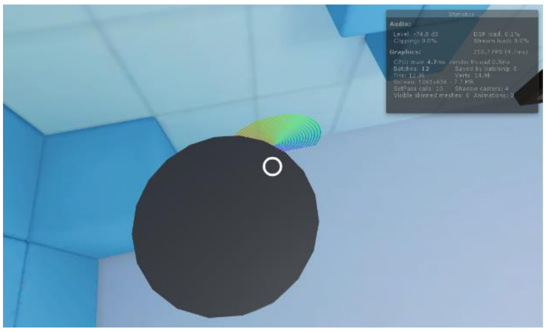
6.3.1.3.2 Test 2



Before moving to the location

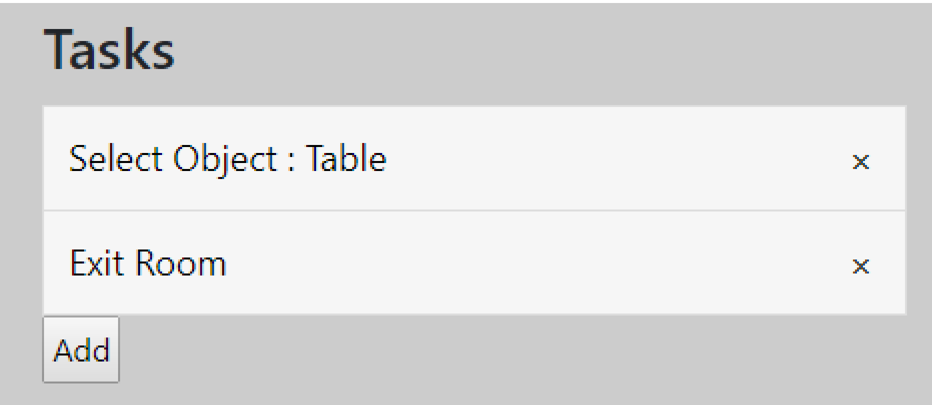


After moving to the location



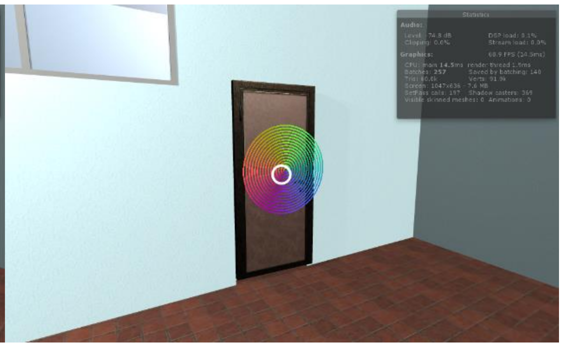
When the user moves to the location indicated with the cylinder, a sound plays to show that they have gone to the location. This confirms the task works as intended but an issue is that the cylinder stays even after they have completed the task, so that would be a bug fix also a cylinder is a poor marker for a waypoint.

6.3.1.3.3 Test 3

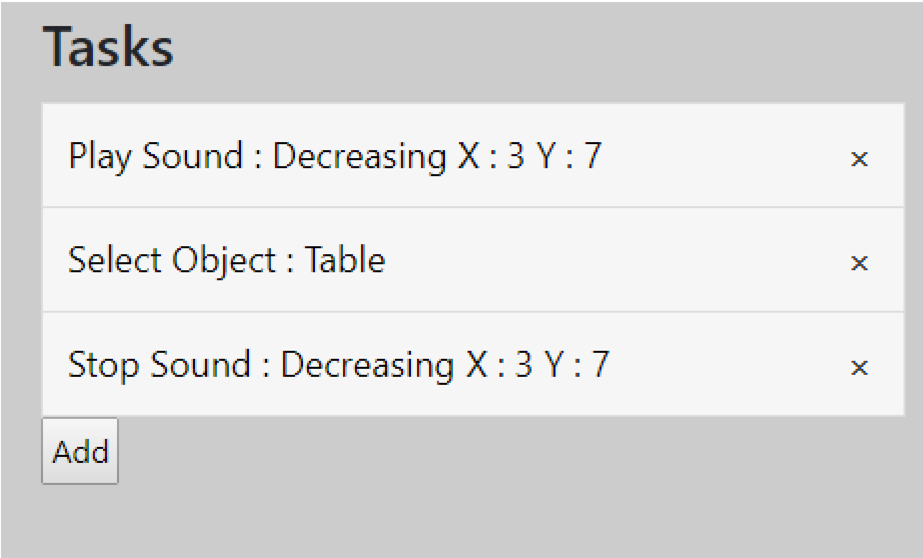


The door is not accessible as soon as the room starts

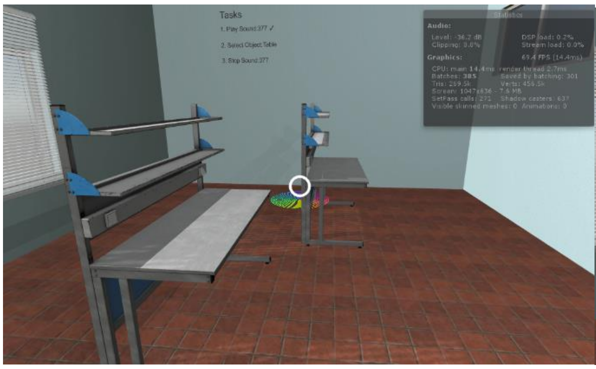
After the object has been selected the door is now active



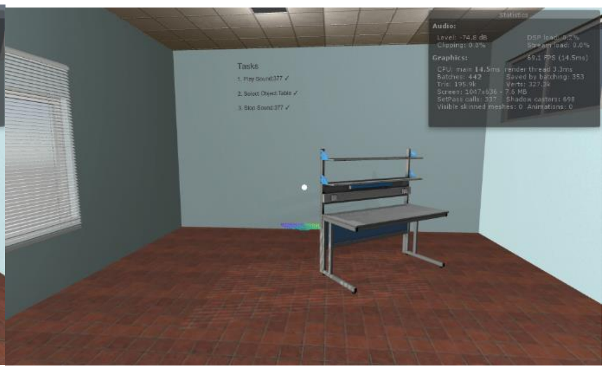
6.3.1.3.4 Test 4



When the game starts there is a sound playing



When the table has been selected, the the sound stops



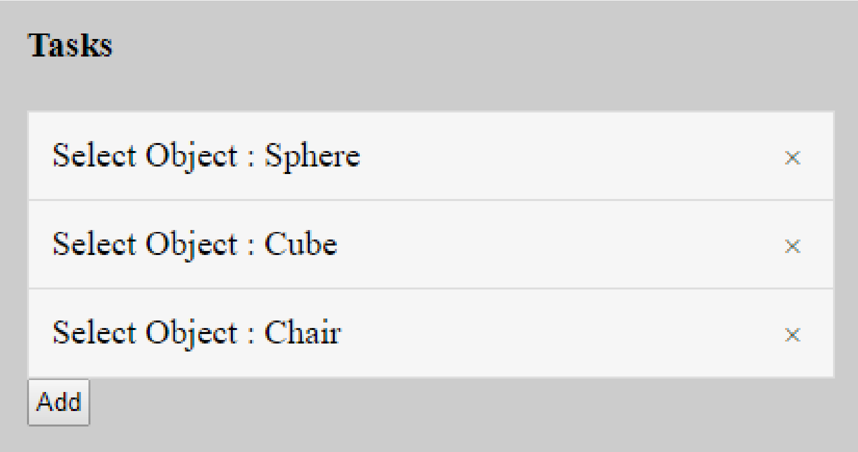
6.3.1.4 Test 4

Goal: The order of the tasks is kept

Test 1: Three select object tasks in a row with specific objects.

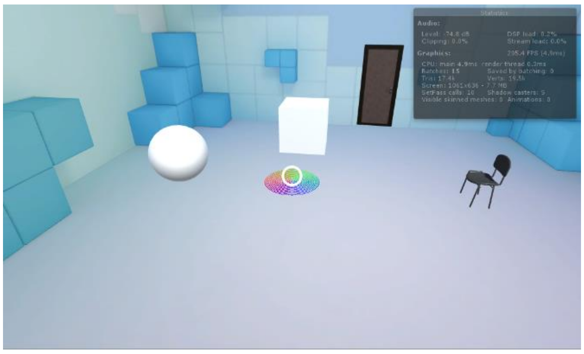
Test 2: Exit room task at the beginning or middle forcing tasks to be incompletable

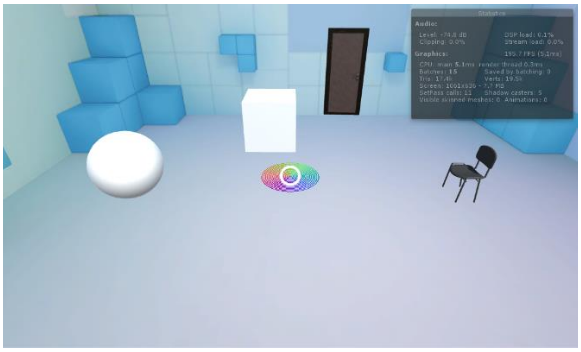
6.3.1.4.1 Test 1



At the start of the test

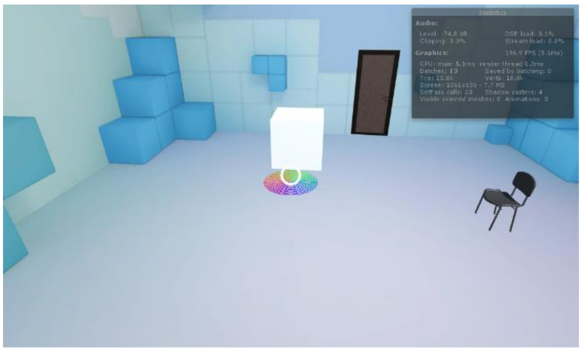
Results of clicking on cube or chair

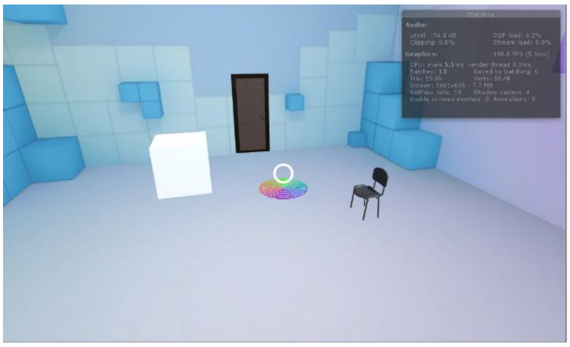




Results of clicking on the sphere

Result of clicking on the chair

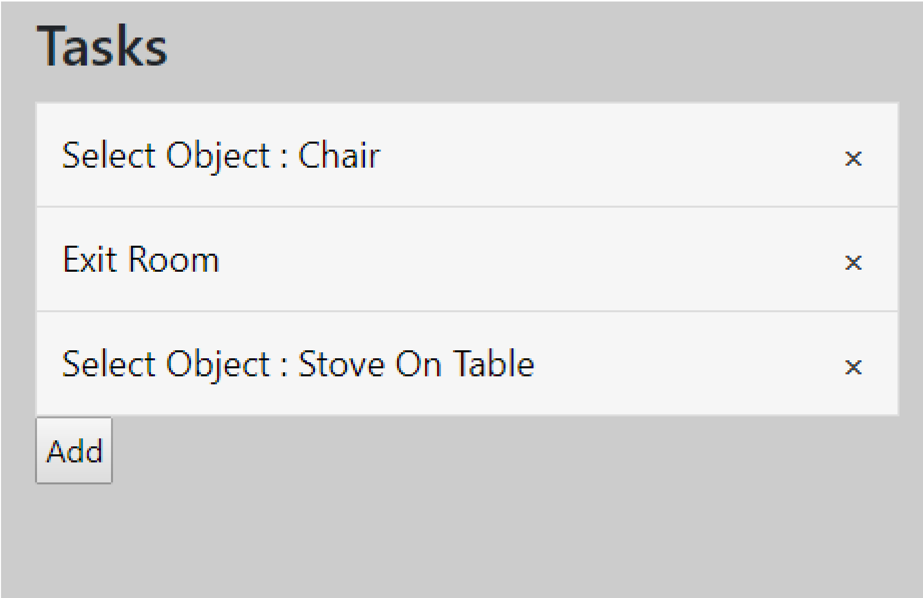




Results of clicking cube

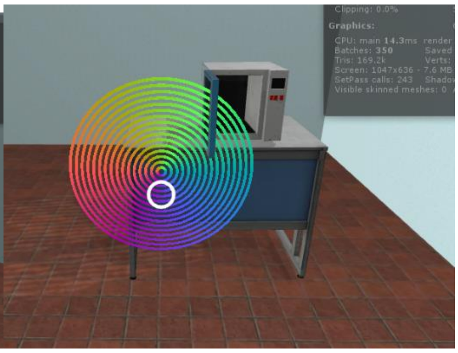
Result of clicking chair

6.3.1.4.2 Test 2

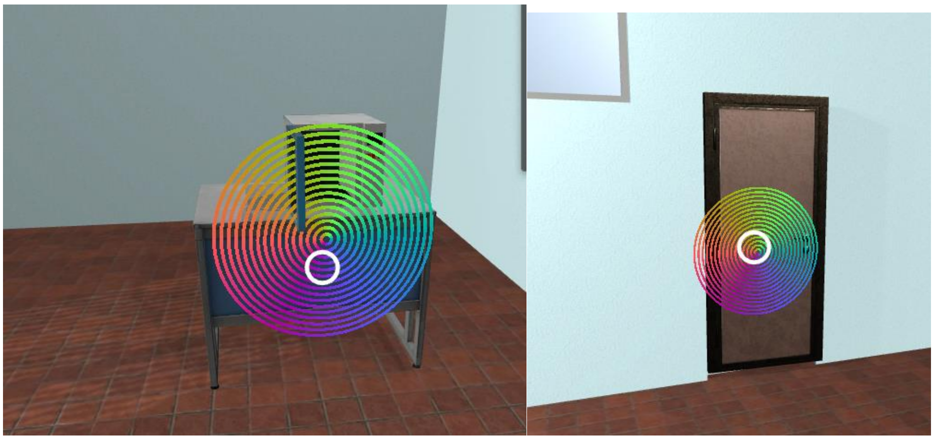


In this test, Selecting the stove should be impossible.

The door is not selectable at the start The stove is also unselectable



After the chair has been selected the stove is selectable and the door is selectable, which is an error in the code.



7 CONCLUSION

In the final section of this report, the success of the project will be evaluated. The evaluation will be done by restating the goals of the project and analysing whether they have been achieved. Leading to looking towards future goals.

7.1 SUCCESS CRITERIA

The success criteria of the project were first defined at the start of this report, in the aims which will be restated here. "To create a tool that will allow academics to easily create their own VR applications to be used in learning environments."

The objectives that must be done to achieve this goal is.

6. Background research into the area
7. Designing the framework of the deliverable
8. Design the online authoring tool, database and mobile application
9. Implement the designs of the three deliverables.
10. Test and evaluate all the deliverables

To go more in-depth on each section. Background research needs to be done in plenty of areas these being VR training tools currently on the market, VR in general, Creation of authoring tools and Creation of VR environments.

Designing the framework is about working how the different parts of the project will interact with each other and deciding the form in which each part will come to represent.

Designing the authoring tool must be done with some objectives in mind, these being to keep the interface simple for the users, making it fit for purpose, allowing plenty of customizability and freedom within the tool. The database must be built to allow smooth interaction between all parts of the project. Lastly, the mobile application needs to be optimised for mobile devices and be able to interpret the authoring tool successfully.

7.2 EVALUATING SUCCESS

7.2.1 Background research

Background research was done into many areas that the project could potentially be interested in pursuing when looking at VR training. Knowledge was gained in previous attempts at doing similar projects. The common thread was that VR training is more engaging and has a higher propensity for learning than traditional training techniques. The research about VR training can be seen in section 3.1.1; the more in-depth look at Health and safety in VR can be seen at 0 and investigate online training can be seen at 3.1.3. Lastly, a section looking at what the Loughborough Health and Safety has requested from the project was described in section 3.1.6

Research into VR practices was done to learn more about the area, to allow for more intelligent decisions in making environments and the mobile application. The research was done into the limitation of VR as a field. This helps to limit the scope of the project and not fall into any of the traps that VR development has this is found in section **Error! Reference source not found.** Motion sickness is the most common complaint when it comes to VR, thus learning how to combat it as much as possible is essential, section **Error! Reference source not found.**. How the user moves in the mobile application is very important as it affects a lot of the design decisions that go into creating an environment because of this research was done into which movement options are available and which would be best suited to this application, section 3.1.4.

Research into correct environment design was done to give knowledge into how to correctly design the pieces that the user will have used to create their environments in the application this can be seen in section **Error! Reference source not found.**

Lastly, research has been done into how to create an authoring tool that the user will enjoy using. This was done by first looking at the history of modding tools and how they have changed over the years. This section ends with looking at many examples of successfully implemented authoring tools, they were analysed, and the right parts that can be implemented into the authoring tool of the project were taken out. Found in section 3.1.5.

7.2.2 Designing of the framework

The overall design of the entire project was done in section 4.1 Inside of which discussion of the multiple ways that the project can be implemented. The decisions were about the entire project. An example of this is the form in which the authoring tool will take place as there was not a single way that It could be implemented.

7.2.3 Design of the deliverables

Design of the authoring tool was done in section 4.2, in this section, all the possible options in how the authoring tool could be designed are considered and compared. A lot of the decisions are made regarding the end users of the project and knowledge learned earlier in the background research phase.

Design of the database is done in section 4.3. Database designing is imperative to correctly functioning and flexible system. The original designs of the database will over time as more features are added or removed will change. However, having a design to base of allows for the other sections of the project to be implemented with confidence that no significant changes will happen.

Design of the Mobile application is done in section 4.4; the mobile application is a bare-bones application for example if it did not have a connection to the internet then nothing will be shown. The design of the mobile application was mainly in the scripts that it was running and how effectively it can translate the authoring tool to the VR environment.

7.2.4 Implementation of the designs

In section 5.1.1, all the features of the project that were seen to be critical to the project working as intended were talked about here with the challenges that were faced when implementing them all. Implementing these were the focus whenever work on the deliverable was done.

In section 5.1.2 features that were deemed to be optional to the project were described and shown to be implemented in this section. Again, along with any challenges that were faced in implementing them.

In section 5.1.3 features that were not implemented are described along with reasons for why they were not.

Section 5.2 gives an in-depth breakdown of the communication channels between the three sections of the project. Lastly, section 5.3 describes and details all the data structures used in the project.

7.2.5 Evaluation of the deliverable

To evaluate whether the deliverable was of the standard that was expected, a series of tests were conducted. The first was requirements testing done first hand to see if the deliverable was able to do all that was required of it. The plan of which is in section 5.2.1 and the results can be seen in 5.3.1. another form of testing was done this as well this being alpha testing. This can be seen in section 5.2.2 and results in 5.3.2

7.2.6 Overall Success

Looking at all the previous paragraphs in this section and the outcome, I believe the project to be successful in achieving its goals. The deliverable is extensive and versatile, and the way that it has been coded allows for more objects and different types of rooms to be added without any issue. The issues of non-preciseness I believe always to be part of the project unless the authoring tool is changed to allow for creation in 3D. There is room for improvement in terms of telling the user what the goals of the room is, as it is hard to know what to do in a room without knowing what is possible. However, overall this is an excellent starting place to allow for more work to be done in the field.

7.3 FUTURE WORK

As good as the project's deliverables there is plenty of room to grow the project due to the flexibility of the tool allows. In this section some of the options that can be done to improve the project even more.

7.3.1 Move to 3D

The most important future work is to convert the authoring tool from its current 2D plane of view to a 3D one, this comes with lots of advantages they are as follows. First and most important is a better view of what the final product will look like. By being able to visualise the room during the building, phase allows the user to make more informed choices on how to build the room. The extra dimension allows for more preciseness in where objects can be placed as it is no longer needed to visualise the room as a bunch of squares; instead, it can be represented as the room. The next advantage is that the Y-axis can be used. Now it is possible to have objects be placed in the air, but it is tough to use. With the 3D interface, it becomes a lot easier. Another is allowing objects to be put on top of each other which now is not possible due to only one object being allowed in each cube going 3d fixes this. Object sizes, which now all objects are the same size as they all fill the same grid square on the tool but in reality, this is not the case a table is not the same size as a fire extinguisher going 3D will allow the objects to be the size that they are. By being a replication of the final product, it would not be a leap to allow the user to playtest the room before they push it to the database.

The disadvantage of doing this is that it will no longer be on a website as the way that will be most easy for the implementation of this would be through unity. Creating a pc program which allows all of this. So, the tool will not be as easy to use, but the added power and improvement in results are worth it. The ease of use can be offset with inbuilt tutorials on the program.

7.3.2 Unlimited amount of rooms

The current deliverable allows the user to make 1, 2 or 3 rooms. This works for testing and the current setup, but for any more complicated scene it would be good to have the ability to create even more rooms.

Implementing this feature is possible, a change to the way that cookies are used on the website will have to be done. As currently, each room is a separate cookie, by knowing that there are only three rooms pre-making the cookies is an option that is used. To allow for unlimited rooms instead of having a cookie per room, one overall cookie for the entire scene will have to be used.

On the database side of things, nothing will have to change as the current database, and mobile deliverable has been set up to allow for the updating of rooms which is like creating more than three rooms as previous rooms are deleted. This would be an easy new feature to add, working how to store the data on the authoring tool correctly would be the challenge.

7.3.3 Upload the users' objects or sounds

The current setup allows the user to pick from only a small set of objects that can be used on the authoring tool. There is no option to allow the user to create their own object and upload it to the tool and then see it in the VR environment.

The feature would be great to add but the feasibility is what holds it back. The reason this would not be possible is based on how objects currently work in the deliverable; all objects are already loaded into the mobile application and all the authoring tool does is say which preloaded objects the user wants and where. Adding new objects means adding more preloaded objects to the application which cannot be done without recompiling the whole app.

The same is true for sounds and tasks; they are all precompiled in the app so adding new ones is not possible as the app currently is. There may be a way to allow for this to be possible some more research into the matter would have to be done.

7.3.4 Better looking menu

The menu within the mobile application is currently adequate; it is not very comprehensive, but for the current revision of the project it works for what is required. To add to the professionalism and quality of the deliverable, a better menu is required. The new menu can add many new features like the quality of life settings, search options and a tutorial for new users.

7.3.5 The objective screen upon entering the room

When a user enters a scene, they are just dropped in without knowing what to do. The adding of the screen which pops up when the user enters the room giving them an initial objective to do within the scene. Adding this pop system would be simple to do on the mobile application but finding a way to add it intuitively to the authoring tool is a lot harder as there would have to be a new side menu or something different as it doesn't seamlessly fit into the tool.

7.3.6 Account system

Currently, all users of the authoring tool can see all other scenes and can edit all other scenes and play all other scenes. This was done as it simple for development purposes as there was only ever one user. If the deliverable were to be used by more then one person which would be a hope, then an account system would be required so people can only edit their scenes without others being able to. Still allowing everyone to play all scene is still a good idea unless the user want to make their scene private.

This feature would mean some changes to the database as it would be required to hold sensitive data now. Changes stated in 7.3.4 would allow for this to become easier as there would now be a massive influx of potential room so having a way of search through them would be helpful.

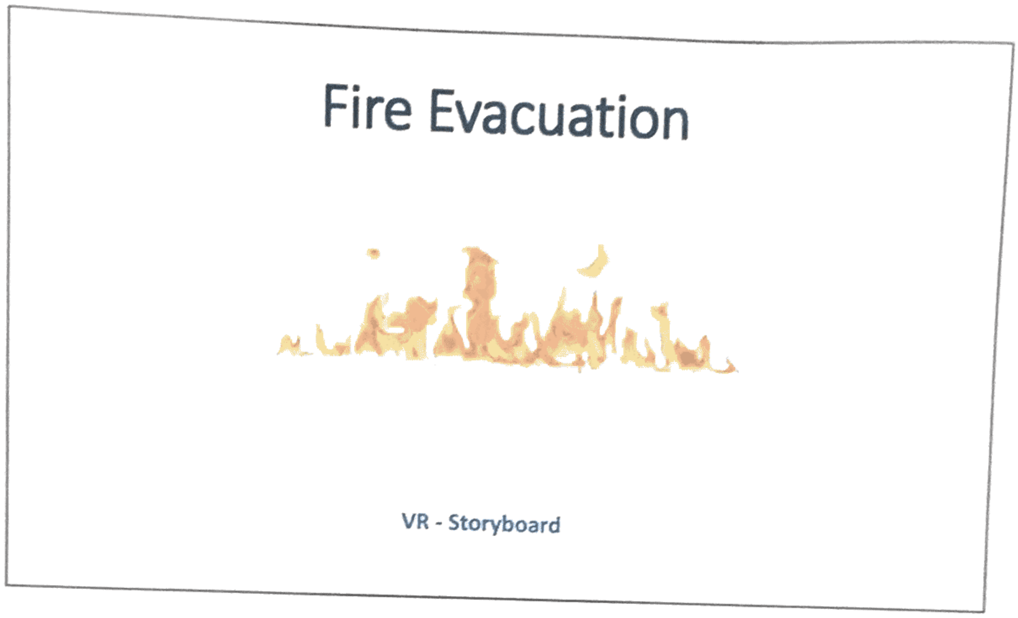
7.3.7 Flow chart tasks

Currently, tasks are linear; tasks are done one after another with no ability to split or create multiple paths. As the project currently is, this is not required, but this ability would allow for a lot more flexibility in creating of scenes.

The best way this can be implemented is using a flowchart system which the user interacts with to create tasks for the room, on this chart tasks, flow together and can have choices which allow them to split off and go on different task routes entirely. Allow for more flexibility within the authoring tool. The implementation of this would be difficult but not impossible as creating a flowchart system on the internet has been done on another website before, but a parser is required to convert the flowchart into a storable data then that data needs to be rebuilt in the mobile application which can be difficult.

8 APPENDIXES

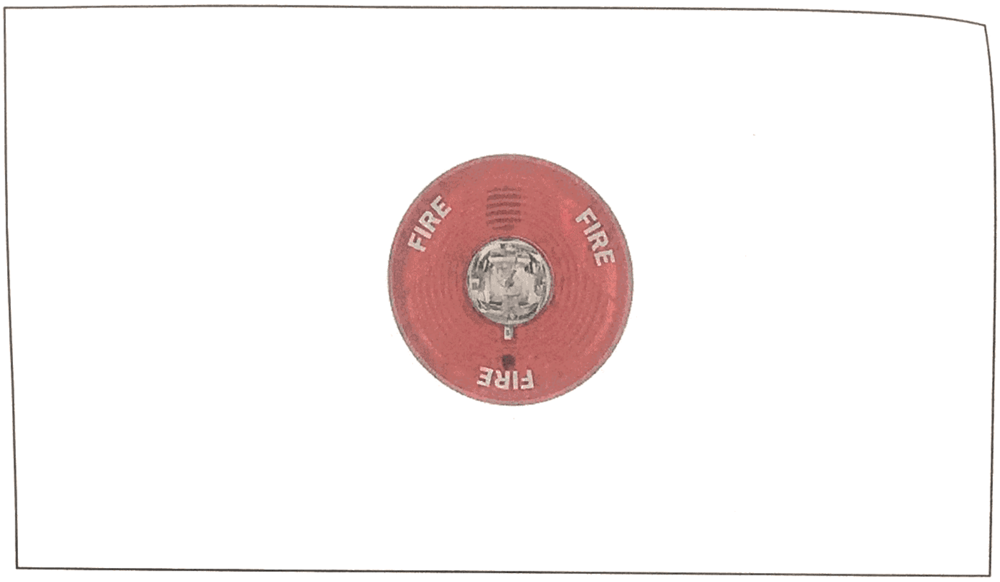
8.1 HEALTH AND SAFETY STORYBOARD



Holding room – internal room within Hazlerigg – Oak Lounge reception area – focus on 3D model in waiting room?
Synched start

VR starts with subject sat at own desk looking across at busy office

Opportunity to sit at desk with 360 view of people working in the office



Fire alarm sounds and continues to ring

As fire marshal the first thing you need to do is put on your hi vis vest – this could be on the back of your chair, in your drawer or on the back of a door

This needs to be an action for the fire marshal to complete before being allowed to move on to the next action

Action around the office needs to be people beginning to get up and make their way out of the building, or just sitting around wondering whether it is a test – some people will be moving, some won't

5

Once hi viz is put on next action will be to start sweeping the building

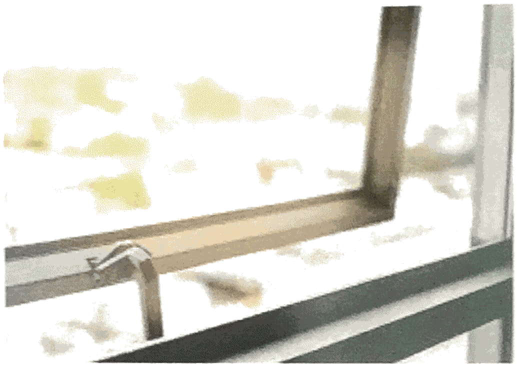
Action will be to direct people to the nearest fire escape and checking rooms, chivvying people along

Alarm sounding loudly in background

6

Next action – sweeping building – entering an office toilet
(Fire marshals sometimes are uncomfortable doing this)

7



Fire marshal carries on with sweeping

Open window comes into view

Action A – closes it, can move on
Action B – leaves it open – fire increases –

8

Carry on with building sweep, come to a door, smoke beginning to come under door

Using back of hand to check heat

Action A – use back of hand, feel heat – move away on to next scenario of calling emergency services and security
Action B – just opens door – fire back draft, huge flames, blank screen, reset to door to repeat action

9

Fire marshal, now established that there is smoke and a fire

Actions:

Call 999 – can we replicate a phone call? – pertinent information required

Call Security

10

Fighting fires

Have all fire extinguishers available and a fire

Selection of the correct extinguisher for the fire being faced

Not to make this too long - option to use two different types of extinguisher on fire from a selection of all available types

Show how to correctly use – checking tag and gauge

The effects of using the wrong extinguisher on the fire

Fire growing/spreading as a result etc

Need to emphasise this is the last line of defence – only used when safe to do so and trained

11

Continuing evacuation

Come across blocked exit

Action A – clear blockage and lead evacuation safely

Action B – use alternative escape route

Alarm sounding, smoke filling area

12

https://www.youtube.com/watch?v=n3nXk5I2gGw

Alarm sounding
Smoke filling escape route
Crowd panicking around you

Action – to control the crowd, calm proceedings and execute a safe and orderly evacuation

13

During evacuation come across a refuge point with mobility impaired user

Actions: check that user has called for assistance

Enable evacuation of person if safe to do so

14

Exit building and lead everyone to the fire assembly point

360 shot of burning building in background

15

Handover of information to fire service/fire co-ordinator

What information – perhaps a white board of options that could be chosen of the pertinent information required?

Ie,

Location of fire
People remaining in building
Person at refuge point

Plus other random options

Burning building in background, fire fighters going about their business

16

End screen shot with what to do next

Book on fire extinguisher user course

Carry out routine checks of emergency routes

Check fire fighting equipment

17

9 REFERENCES

Andreatta, P. B. et al., 2010. Virtual reality triage training provides a viable solution for disaster-preparedness. *Academic emergency medicine,* 17(8), pp. 870-876.

Beal, V., 2019. *mod.html.* [Online]
Available at: https://www.webopedia.com/TERM/M/mod.html
[Accessed 22 04 2019].

Bowman, D. A. & Hodges, L. F., 1997. An Evaluation of Techniques for Grabbing and Manipulating Remote Objects in Immersive Virtual Environments. *SI3D,* Volume 97, pp. 35--38.

Brooks, J. O. et al., 2010. Simulator sickness during driving simulation studies. *Accident Analysis & Prevention,* 42(3), pp. 788-796.

EON Reality, 2019. *Creator-AVR.* [Online]
Available at: https://www.eonreality.com/platform/creator-avr/
[Accessed 16 04 2019].

Gallagher, et al., 1999. Virtual reality training in laparoscopic surgery: a preliminary assessment of minimally invasive surgical trainer virtual reality (MIST VR). *Endoscopy,* 31(04), pp. 310--313.

Halo Alpha, 2019. *Forge.* [Online]
Available at: https://halo.fandom.com/wiki/Forge
[Accessed 22 04 2019].

Hsu, E. B. et al., 2013. State of virtual reality based disaster preparedness and response training. *PLoS currents,* Volume 5.

Jerald, J., 2015. *The VR book: Human-centered design for virtual reality.* s.l.:Morgan & Claypool.

Kanumuri, P. et al., 2008. Virtual Reality and Computer-Enhanced Training Devices Equally Improve Laparoscopic Surgical Skill in Novices. *JSLS : Journal of the Society of Laparoendoscopic Surgeons / Society of Laparoendoscopic Surgeons,* Volume 12, pp. 219-26.

Kolasinski, E. M., 1995. *Simulator Sickness in Virtual Environments.* Alexandria VA: Army research Inst for the behavioral and social sciences.

Lackner, J. R., 2014. Motion sickness: more than nausea and vomiting. *Experimental brain research,* 232(8), pp. 1293-2510.

Lang, B., 2015. *visionary-vr-filmmakings-most-fundamental-concept-virtual-reality-storytelling-frame.* [Online]
Available at: visionary-vr-filmmakings-most-fundamental-concept-virtual-reality-storytelling-frame/
[Accessed 22 04 2019].

Lawson, B. D., 2014. Motion sickness symptomatology and origins. In: *Handbook of Virtual Environments.* s.l.:CRC press, pp. 536--605.

Lin, J. J., Abi-Rached, H. & Lahav, M., 2004. Virtual guiding avatar: An effective procedure to reduce simulator sickness in virtual environments. In: *Proceedings of the SIGCHI conference on Human factors in computing systems.* s.l.:ACM, pp. 719-726.

McComas, J., MacKay, M. & Pivik, J., 2002. Effectiveness of virtual reality for teaching pedestrian safety. *CyberPsychology & Behavior,* 5(3), pp. 185-190.

modmoddermodding, 2011. *History of Modding.* [Online]
Available at: https://mediaindustries1.wordpress.com/modmoddermodding/history-of-modding/
[Accessed 22 04 2019].

Prothero, J. D. & Parker, D. E., 2003. A Unified Approach to Presence and Motion Sickness. *Virtual and adaptive environments: Applications, implications, and human performance issues,* p. 47.

Razzaque, S., Kohn, Z. & Whitton, M. C., 2005. *Redirected walking.* s.l.:Citeseer.

Reason, J. T. & Brand, J. J., 1975. *Motion sickness..* s.l.:Academic press.

Riccio, G. E. & Stoffregen, T. A., 1991. An ecological theory of motion sickness and postural instability. *Ecological psychology,* 3(3), pp. 195-240.

Slater, M. & Wilbur, S., 1997. A framework for immersive virtual environments (FIVE): Speculations on the role of presence in virtual environments. *Presence: Teleoperators & Virtual Environments,* 6(6), pp. 603--616.

Smid, A., 2017. Comparison of unity and unreal engine. *Czech Technical University in Prague.*

Squelch, A., 2001. Virtual reality for mine safety training in South Africa. *Journal of the Southern African Institute of Mining and Metallurgy,* 101(4), pp. 209-216.

Stanney, K. M., Graeber, D. & Kennedy, R., 2006. Virtual environment usage protocols. *Handbook of standards and guidelines in ergonomics and human factors,* pp. 381-397.

Usoh, M. et al., 1999. Walking> walking-in-place> flying, in virtual environments. In: *Proceedings of the 26th annual conference on Computer graphics and interactive techniques.* s.l.:ACM Press/Addison-Wesley Publishing Co., pp. 359--364.

wikipedia, 2019. *Super Mario Maker.* [Online]
Available at: https://en.wikipedia.org/wiki/Super_Mario_Maker
[Accessed 22 04 2019].