A Virtual Reality Program that simulates fire emergencies in accurate models of real-life Newcastle University buildings.

Mungo Blyth (Student ID: 170471699)

BSc Computing Science, August 2020

Supervisor: Giacomo Bergami

Word Count: 10,318

Declaration

"I declare that this dissertation represents my own work except where otherwise stated."

Acknowledgements

I would like to thank my family, especially my parents, for always supporting me throughout my education.

Abstract

Using Virtual Reality programs as an educational tool is a growing industry. Harnessing VR simulations for training and educational purposes can achieve an immersive experience in a controlled but versatile environment that can provide benefits to the effectiveness of the training over traditional methods. This dissertation project aims to model environments that are familiar to Newcastle University staff and students in Virtual Reality, and then use these as a foundation on which to simulate fire training scenarios. The main goal of this is to provide a training experience that improves the reactions of users in the event of emergency fire situations.

Table of Contents

Introduction

1.1 Introduction

This introductory chapter aims to give the reader a firm grasp on the material that this dissertation will cover, as well as to help them understand the main goal I wanted to achieve with this project. The outline serves as a guide to the different chapters of the paper. Followed by this is the statement of the aims and objectives of the project, and the requirements the system would need in order for these objectives to be implemented correctly. I then discuss the limitations experienced during the project, and the changes I made to account for these, ending with a brief summary of the chapter.

1.2 Outline of the Dissertation

Chapter 1 – Introduction

The introduction is designed to help the reader understand the goal of the project, to state my aims and objectives, as well as discuss the limitations and changes made by the project.

Chapter 2 – Background

This chapter gives an analysis of background material, research publications and projects that are relevant to my project. There is an overview of research projects that achieve a similar goal, and an assessment of the commercial landscape.

Chapter 3 – Developing the Program

Chapter 3 explores the process by which I created the program. I discuss the design of the system from a software engineering aspect, and how each element was designed to meet the aims and specification of the project, as well as a brief on the different technologies that would be used in the system. A high level description of the development process follows, with explanations of the choices I made throughout implementing the different aspects of the system, and how these implementations related to the original design.

Chapter 4 – Testing, Results and Analysis

In this chapter I display the testing I performed on my software and the results I've produced. I analyse and evaluate the data against the objectives of the project.

Chapter 5 – Conclusion and Evaluation

For the conclusion of my work, I reflect upon the project as a whole, discussing what I have learned. I evaluate the end result of my project against the initial aims and objectives with which I set out, and discuss the software engineering aspects of the development.

1.3 Motivations

As VR hardware becomes more advanced, and therefore immersive, the software will inevitably have increased and more powerful uses, such as a greater variety, efficiency and effectiveness in programs that help to train people in some role, such as engineering, safety, construction, or medicine. I chose to create a fire training program because of the importance of fire safety in modern architecture and risk assessment. The importance is relevant because modelling of fire safety and risk assessment is increasingly computer-based. I then decided to use the Urban Sciences Building as the source of my modelling because the data about the building was readily available to me.

1.4 Aims and Objectives

Aim:

The overall aim of this project was to build models of Newcastle University buildings in a virtual reality environment, to introduce fire hazards and scenarios to these environments, and to develop training missions which could then be tested on a number of individuals. The data from these experiments could then be analysed in order to evaluate the effectiveness of VR training for individuals and to give insight into existing levels of fire safety.

This was to be accomplished through the following objectives:

1. Create models of University buildings, specifically the Urban Sciences Building, that are playable levels and accurate to a good degree.

   I aimed to create the Urban Sciences Building as the main level, and to create one other University building in order to compare and contrast how different design and architecture had an impact on the results of tests on users.

2. Design training that simulates several specific fire hazard scenarios in these environments, and clear objectives to solve them.

   The original scenarios that I wanted to achieve in simulation were the use different types of fire extinguishers for different fires, and a search-and-rescue mission to find AI who had not evacuated the building.

3. Implement VR compatibility in order to allow for this training to be performed with a Head Mounted Device and hand controllers.

Full VR compatibility with the main headsets that are currently available, such as the Oculus and the HTC Vive.

4. Combine these aspects within the game engine to create an immersive VR Fire Safety Training Program that can improve the reactions of users fire emergency scenarios

   Utilising these different aspects in conjunction will allow for an immersive training experience uniquely adapted to environments that are familiar to the users of the program.

5. Evaluate how data from the simulations can be applied to fire safety, and whether user reactions to fire scenarios are improved. In addition, evaluate the efficiency and effectiveness of the implementation.

   Data from users reactions to the tests could allow insight into the level of emergency fire scenario preparedness, and how VR simulations can improve training for fire scenarios.

1.5 System Requirements

In order to achieve the above objectives, the following features would need to be implemented in the system:

1. The system must have a modelled environment in which the user wearing an HMD may be placed
2. The system must have working integration with the XR Framework, and be compatible with a range of VR hardware.
3. The user must be able to input commands which control the avatar within the simulation
4. The system must enable the user to complete objectives through the use of interactable objects and a User Interface (UI)

1.6 Limitations and Changes

Some of the limitations experienced during this project were due to the University unfortunately being closed a result of the COVID-19 pandemic. These include:

1. The loss of the VR device to be used for testing my program.

2. I could not visit University buildings in order to help me to model them in Unity

3. I was unable to test the project on volunteers in order to gain data about the simulation.

The other limitations encountered were:

4. The implementation of the smoke system in unison with the USB model was not compatible with the training objectives I wanted to achieve.

   The Urban Science building has a very open centre, meaning the dissipation of smoke in most areas of the simulation was not going to allow for any meaningful smoke-related objective for the training missions.

5. The visual aspect of the game environment was limited by the lack of default art assets available.

   Each object in the simulation needs to be given a material and a texture. Realistic materials and textures take graphic design skills to create. Some default assets are available, but they generally did not suit this particular environment. Paid assets are more realistic, but not an option.

These limitations led to the following changes in the strategies for development and the overall aims of the project:

1. The creation of a First-Person character controller to run alongside the XR Rig (Virtual Reality Controller) with Mock HMD settings in order to test my environment.

2. I focused on modelling the Urban Sciences Building as the core of my training environment, using floor plans and video to recreate it.

   I created the USB structure and filled it with the rooms and offices that exist in the building in real life.

3. I implemented AI to generate data for the simulations.

I programmed AI to evacuate the building from different areas and used their data such as time taken to demonstrate how the implementation can provide meaningful results.

4. I focused on evaluating the efficiency and effectiveness of the product.

1.7 Summary

This section has outlined the abstract and high level goals of the project in order to introduce the reader to the dissertation. The way in which these goals translate to the requirements of the system to be used for the project give a technological insight into the foundations of the development of the system. The limitations and changes give a glimpse into the course of development that is discussed in more detail in chapter 3.

2 Background

2.1 Introduction

In this section I detail the research that I conducted into the background areas of my project. I discuss the game engine used for development, and explore the nature of virtual reality and the areas of fire safety which I intended to include in my simulation. Moreover, I compare my project to existing applications, and the viability of this type of project on a commercial level, with an examination of the broader landscape for XR in education and training.

2.2 Research

2.2.1 Virtual Reality

Virtual Reality (VR) is a simulated experience that can be defined as "a three-dimensional, computer generated environment which can be explored and interacted with by a person" [1]. It belongs to a wider spectrum of technology-altered reality, given the umbrella term of XR (Extended Reality). This spectrum, named the Reality-Virtuality Continuum [2], as seen in Figure 2.1, also encompasses Augmented Reality (AR) which is the projection of digital images upon real environments, and Mixed Reality (MR) which is a broad term for a combination of real and digital elements. VR would be considered the far-right of this spectrum.

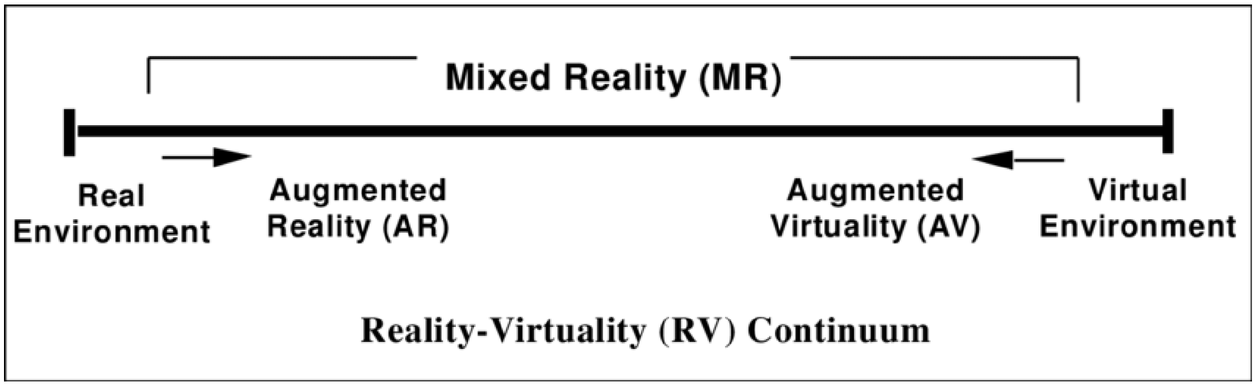


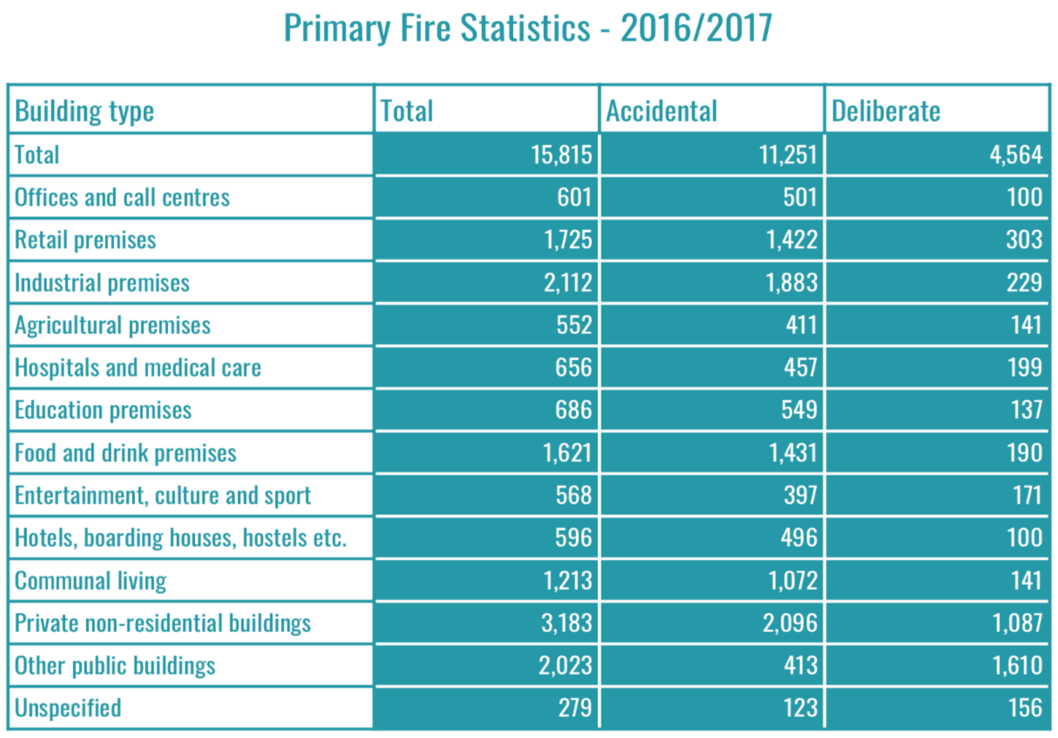
Figure 2.1: Reality-Virtuality Continuum [2]

An individual generally interacts with the environment via a Head Mounted Display (HMD) and a input device, often a haptic controller that mimics hands, used for controlling the locomotion and interactions of their avatar. Most of the popular VR devices, such as the HTC Vive, the Oculus Rift, or the Steam Index, are a combination of an HMD and two hand-held controllers. However, it is not limited to this, as there are also a number of programs that aim to make VR more accessible, such as Google Cardboard which acts as an assemble-at-home HMD with which you can use a smartphone (iPhone or Android) to access VR. Google Cardboard does not have hand-controllers however, and therefore would not have been useful for the simulation.

In terms of virtualization, these different methods of accessing virtual reality have varying capacities. The HMD and controller setup offers the greatest level of virtualization out of these options. As the smartphone options do not have controllers, they offer the least virtualization. Hence the more advanced setup (HMD & Controllers) was opted for use in this project, as it would offer a more immersive and interactable experience.

2.2.2 Fire Safety

In order to properly emulate fire scenarios it was important to understand the different areas of fire safety. Firstly, as seen in Figure 2.2, the risk of a fire scenario is not insignificant. In the UK alone in the 2016-2017 period, there were 15,815 fires total, and 686 of those were on educational premises, such as those modelled in my project.



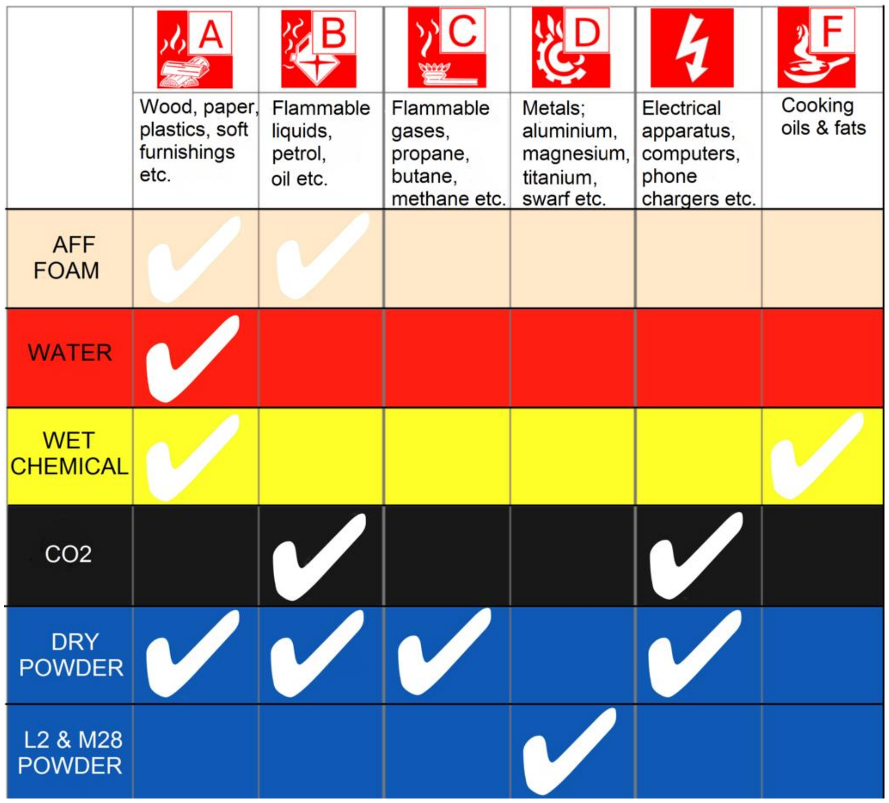
Primary Fire Statistics - 2016/2017

Building type	Total	Accidental	Deliberate
Total	15,815	11,251	4,564
Offices and call centres	601	501	100
Retail premises	1,725	1,422	303
Industrial premises	2,112	1,883	229
Agricultural premises	552	411	141
Hospitals and medical care	656	457	199
Education premises	686	549	137
Food and drink premises	1,621	1,431	190
Entertainment, culture and sport	568	397	171
Hotels, boarding houses, hostels etc.	596	496	100
Communal living	1,213	1,072	141
Private non-residential buildings	3,183	2,096	1,087
Other public buildings	2,023	413	1,610
Unspecified	279	123	156

Figure 2.2: Fire Occurrence Statistics for the UK in 2016-2017 [3]

There are several different fire extinguishers that are designed for different types of fires. Using the wrong type of extinguisher for a fire can have negative consequences.

According to IFSEC Global, “Each year, there are 40,000 fires in the workplace, which can put the lives of workers at risk but using the wrong type of fire extinguisher can also have major consequences”. Furthermore, “Nearly four in 10 (38%) construction workers are using the wrong type of fire extinguisher to deal with electrical fires” [4].



	A	B	C	D		F
	Wood, paper, plastics, soft furnishings etc.	Flammable liquids, petrol, oil etc.	Flammable gases, propane, butane, methane etc.	Metals; aluminium, magnesium, titanium, swarf etc.	Electrical apparatus, computers, phone chargers etc.	Cooking oils & fats
AFF FOAM	✓	✓				
WATER	✓					
WET CHEMICAL	✓					✓
CO2		✓			✓	
DRY POWDER	✓	✓	✓		✓	
L2 & M28 POWDER				✓		

Figure 2.3: The distinct applications of the 6 types of extinguisher [5]

This research determined that it would be important to include more than one type of fire extinguisher in the simulation and train the user to choose the appropriate extinguisher for a fire.

2.2.3 VR in Unity

Unity’s Extended Reality (XR) integration is a framework that allows developers using Unity to implement VR, AR, and MR with ease. By creating a single Software Development Kit (SDK), Unity has created a system that is simple and versatile for both developers and VR providers. Developers using Unity can implement VR functionality and interactions using the XR Interaction Toolkit. This toolkit is linked to the XR Plugin Framework, which contains the XR SDK. The XR SDK allows VR providers to create plugins which can be downloaded by the developer, making their program compatible with that particular style of XR, such as the Oculus or Windows MR. This is illustrated in Figure 2.4.

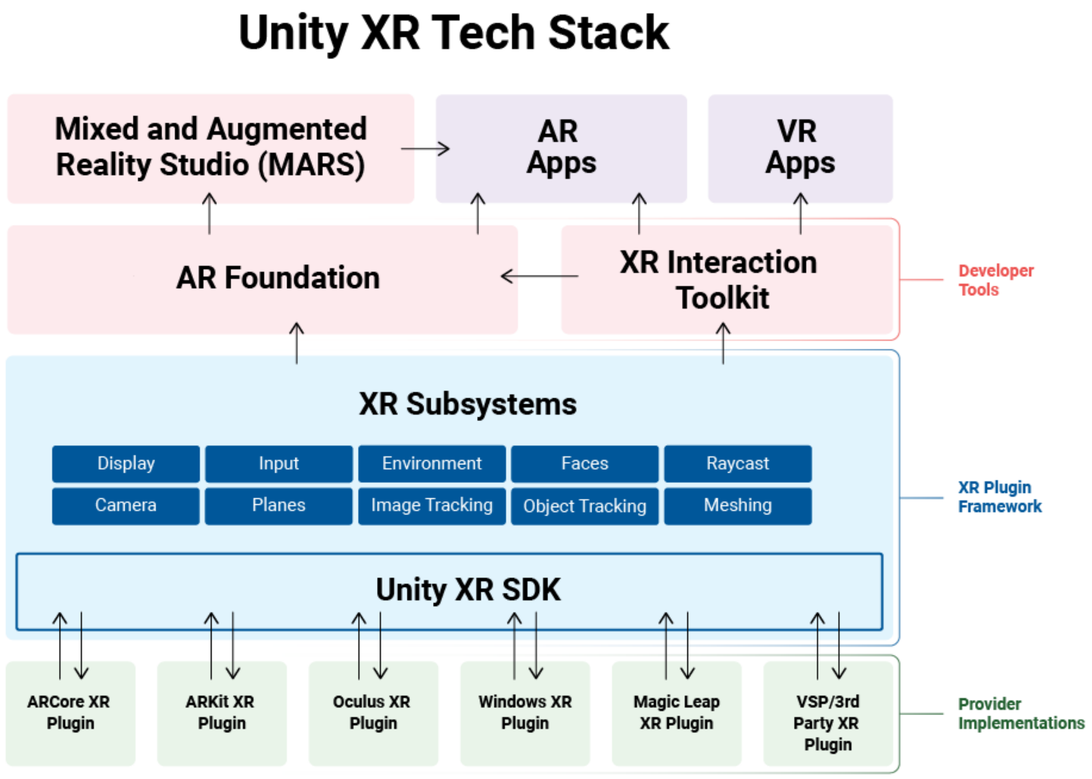
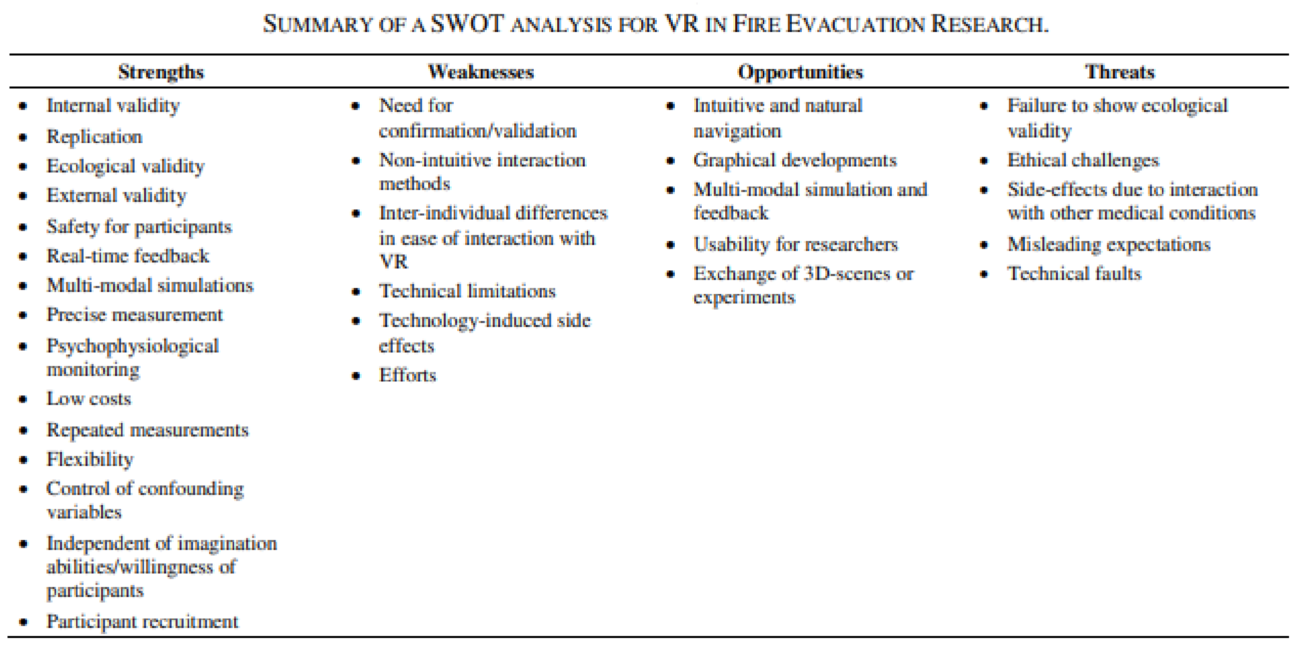


Figure 2.4: The Unity XR Framework [6]

2.2.4 Fire Safety in VR

Virtual Reality for Fire Evacuation Research [7] offers a good oversight into the aspect of an evacuation scenario simulated in VR. The paper praises the opportunities offered by this new medium~~,~~ and analyses the strengths and weaknesses it holds. “Increasingly realistic simulations and other technological advances provide new opportunities for this relatively young method”. A concise summary is presented in Figure 2.5 in the form of a Strength, Weaknesses, Opportunities, and Threats (SWOT) table.



SUMMARY OF A SWOT ANALYSIS FOR VR IN FIRE EVACUATION RESEARCH.

Strengths	Weaknesses	Opportunities	Threats
• Internal validity	• Need for confirmation/validation	• Intuitive and natural navigation	• Failure to show ecological validity
• Replication	• Non-intuitive interaction methods	• Graphical developments	• Ethical challenges
• Ecological validity	• Inter-individual differences in ease of interaction with VR	• Multi-modal simulation and feedback	• Side-effects due to interaction with other medical conditions
• External validity	• Technical limitations	• Usability for researchers	• Misleading expectations
• Safety for participants	• Technology-induced side effects	• Exchange of 3D-scenes or experiments	• Technical faults
• Real-time feedback	• Efforts		
• Multi-modal simulations			
• Precise measurement			
• Psychophysiological monitoring			
• Low costs			
• Repeated measurements			
• Flexibility			
• Control of confounding variables			
• Independent of imagination abilities/willingness of participants			
• Participant recruitment			

Figure 2.5: SWOT Table for Fire Evacuation in VR [7]

The clear limitations of such simulations are best summarised here. “Virtual reality is not reality. Participants will always know that they take part in an artificial situation. It is impossible to generate situations in which participants would risk actual physical harm. Extremely perilous situations may induce effects (e.g., extreme fear) which are not attainable with artificial scenarios, which in turn may affect behaviour” [7].

Comparing the effectiveness of fire extinguisher virtual reality and video training [8] is a study into the effectiveness of the use of virtual reality training on subjects ability to use a fire extinguisher appropriately. The research used a virtual reality program, the fire extinguisher aspect of which is similar to that incorporated by this project, and used it to train a group of participants, whilst a video about using fire extinguishers was shown to a second group of participants. The outcome of the study displayed that “compared to video training, the VR training provided a more effective training result in terms of knowledge acquisition and retention, and self-efficacy.” Furthermore, “Overall, the results identified that the VR-based training tool provided a more effective solution for fire extinguisher training than video training”.

2.2.5 Pathfinding

As the project would be using AI that would have to navigate paths through the modelled building in order to evacuate, taking fire into consideration, it would be important to understand the pathfinding involved, which would take form in the A* Algorithm. “A* (pronounced as "A star") is a computer algorithm that is widely used in pathfinding and graph traversal. The algorithm efficiently plots a walkable path between multiple nodes, or points, on the graph" [9]. The algorithm provides an advantage over basic pathfinding by 'planning ahead', and is an extension of Djikstra's algorithm [10].

2.3 Similar Research

A virtual reality based fire training simulator with smoke hazard assessment capacity is a research paper that focuses on the aspect of evacuation in a VR scenario. The paper has a focus on the mathematics of the spread dynamics of smoke in the event of an indoor fire, and translating this to the virtual environment in order to evaluate the response of trainees in such an event and to assist in teaching them how to minimize smoke hazards [12].

Using a virtual fire extinguisher as a tool for safety training [13] aims to create a realistic fire extinguisher training scenario in virtual reality. The paper highlights the successes of such simulations as a training tool, and also discuss their shortcomings, such as "the lack of noise and weight from real fire extinguishers" [13].

Building a Virtual Reality Fire Training with Unity and HTC Vive [14] is a research project with a similar aim to the above [13]. However, in this research a real life fire extinguisher was used in conjunction with the hand-held remotes, in order to allow for the weight to be understood by the user, a shortcoming of VR noted in [13].

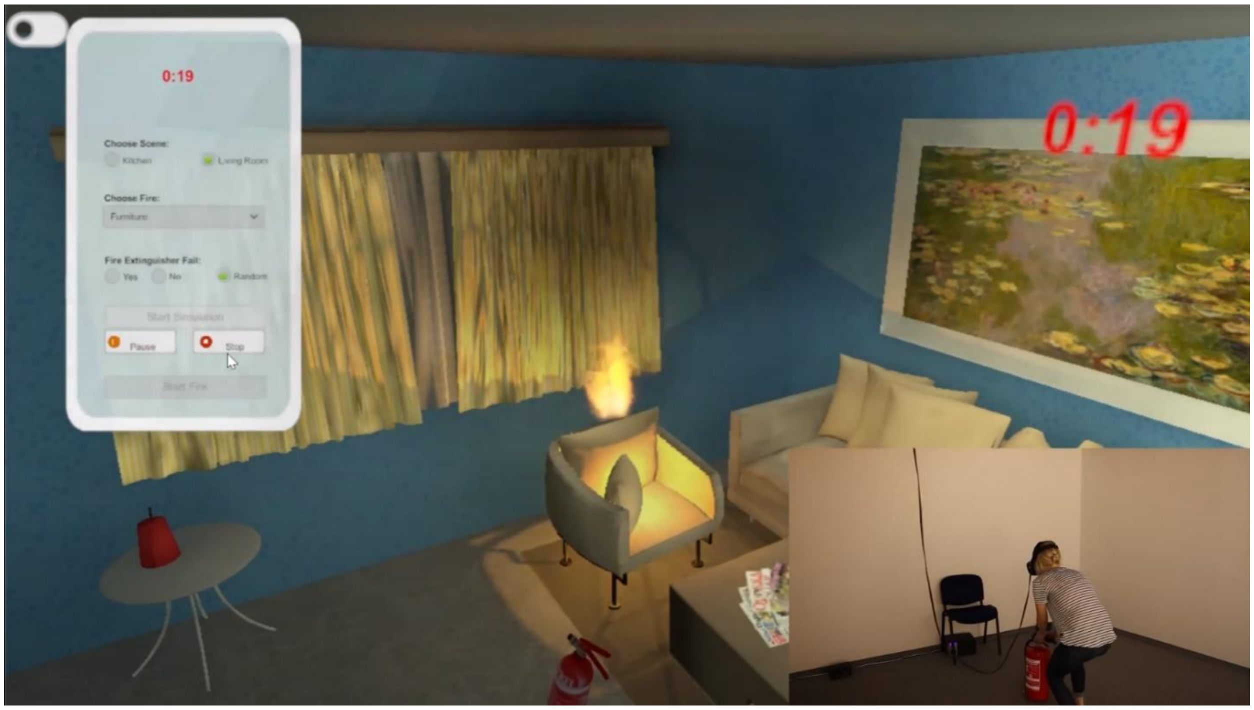


Figure 2.6: A similar VR project where a real fire extinguisher is used [15].

2.4 Commercial Applications

In relation to the above, there are also similar projects that exist to serve a commercial purpose. As the market for XR continues to grow so does the market for VR training. The capacity to enhance training through experiential learning in a safe and controlled environment is being recognised. This is reflected in the forecast $18.8bn market size forecast for AR and VR combined in 2020, seen in Figure 2.7. There are a multitude of companies who offer VR training services specifically tailored to fire emergencies. They have developed their own simulated environments and offer these services to those looking to train their employees.

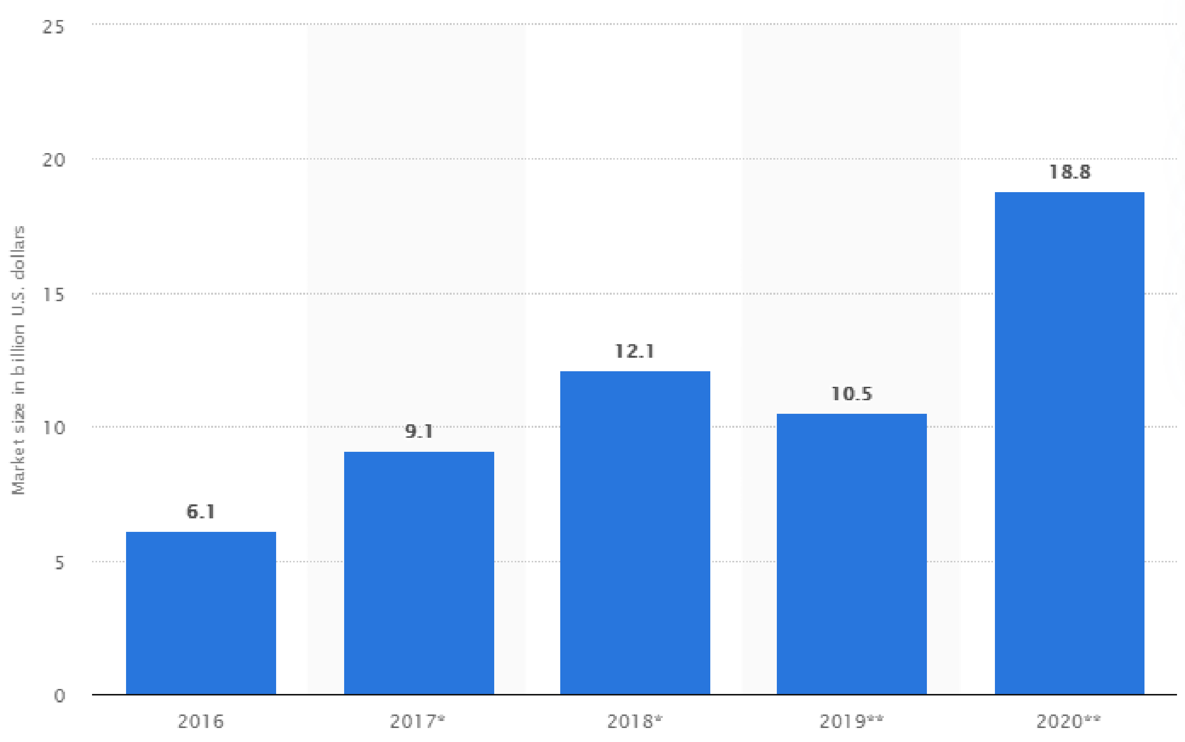


Figure 2.7: VR and AR combined market value (forecasted) [16]

Luminous Group [17], a UK based company, is an example. They develop VR training solutions for industries, such as oil and gas, including fire safety training. Jasoren [18] is a company that develops custom solutions for companies, working with 3D models and XR to provide customised training and education services to companies, predominately VR and AR experiences. One such project is a simulation of a fire emergency in a subway. The purpose of the project was “Training in the correct sequence of actions in an emergency is the primary key to saving lives and preventing a catastrophe. Virtual reality makes it possible to recreate a dangerous situation and practice actions until it becomes second nature without an immediate threat to the learner and the cost of material resources”.

These companies generally offer a product that consists of a generic office environment and a focus on the fire extinguisher aspect of fire safety. In contrast, my project has an environment that will be familiar to the user and has a variety of implemented training scenarios beyond the use of fire extinguishers. The advantages of this include a better understanding of how to respond to a fire in the specific environment and learning a wider variety of techniques on how to respond to an emergency. The disadvantage is that these commercial products likely have superior training regarding the fire extinguisher specifically. However, this is not a problem with regard to this dissertation. Whilst the aims are alike, the implementations do not need to be on par in order to achieve the aims of this project, and the more polished aspects of these professional products should be considered more as potential future work.

The existence of these companies as well as the statistics shown above indicate that there is a growing market for the development of VR training and education tools. This is supported by research from Huawei [19].

2.5 Summary

This section gives the reader an understanding of the core concepts of virtual reality and fire safety involved in the design and technology of the system that the project aims to develop. After discussing the topics individually, I demonstrate how the two elements work in unison when applied to this project, partially through the use of research into the field of the use of VR for fire safety training. The research projects that follow show successes in this field with regard to the potential of retention of skills and increased effectiveness of training. In addition, the commercial landscape for these VR training products is seen to look promising.

3 Developing the Program

3.1 Introduction

The following chapter details the development of the project. It begins with a software engineering outlook on the development, discussing the design, architecture, and planning of the system, as well as the technologies that would be used. This is subsequently followed by a high-level explanation of the different elements developed to implement the aim and objectives discussed in section 1.4. The three key areas of development to note are the modelling (3.5), XR and Test controllers (3.6), and training implementation (3.7).

3.2 System Architecture

Figure 3.1 displays a high-level view of the program that envelops the key elements, each of which will be discussed at length through the next chapter. The diagram illustrates how these elements interact with one another in order to produce the main simulation.

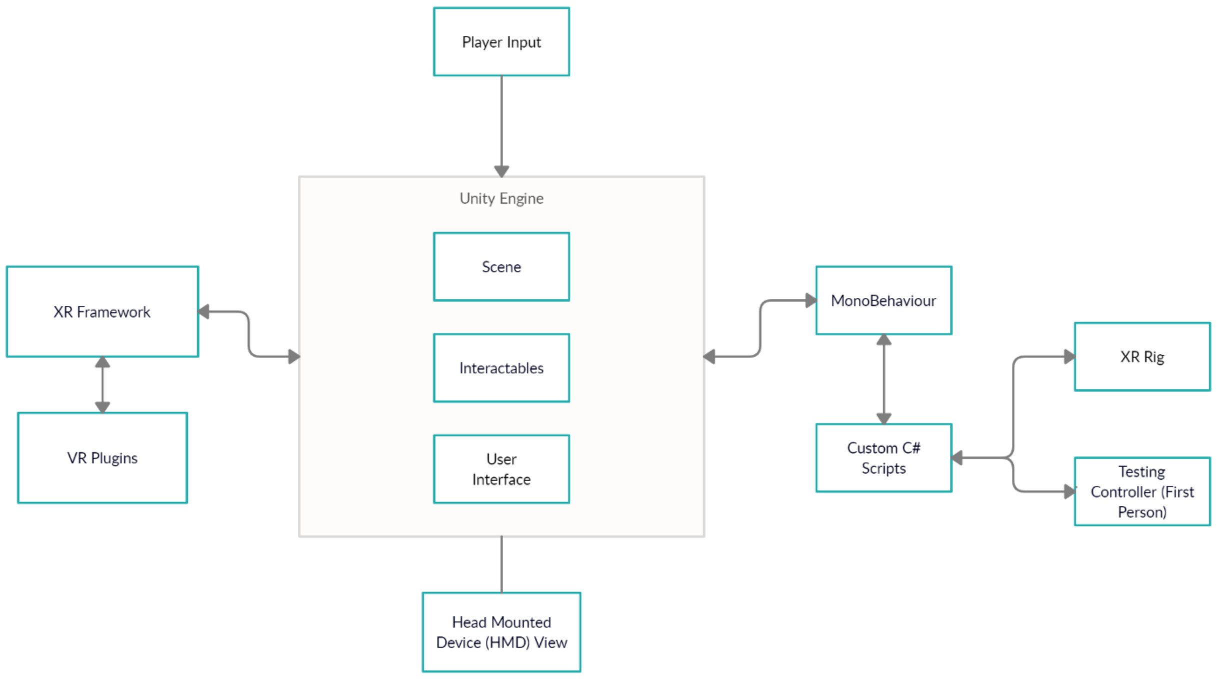


Figure 3.1: High Level System Architecture UML Diagram

Incorporated here are the major components that represent the implementation of the key objectives of the project:

- XR Compatibility
- A User Interface
- User Controls (XR Rig)

- A modelled environment (Scene)
- Interactions for the user to complete objectives (Interactables)

MonoBehaviour is the Unity class through which all custom scripts must interact. It controls the behaviour of all objects implemented in the game. It is through this class that I have scripted the first-person controller used for testing, as well as the XR Rig designed for the training.

The model of the USB, the interactable items, the particle systems, and the AI are all created using the Unity engine, so these are shown encompassed in Unity in the diagram.

The Unity XR Framework interacts with plugins that are developed by the companies who create VR devices. For example, Oculus develop the Oculus XR Plugin for Unity. Therefore these plugins are represented externally in the diagram. These plugins allow Unity to be compatible with a variety of devices, whilst using generic outputs.

3.3 Software Development Model

As with any programming project, it was important to analyse the software engineering aspect in order to ensure a well-planned and thought out cycle of development. The Software Development Life Cycle (SDLC) is described as “a framework that defines the steps involved in the development of software at each phase. It covers the detailed plan for building, deploying and maintaining the software” [21].

The SDLC contains a number of different models that can be followed to ensure a planned process. After much evaluation of the different models, the Modified Waterfall

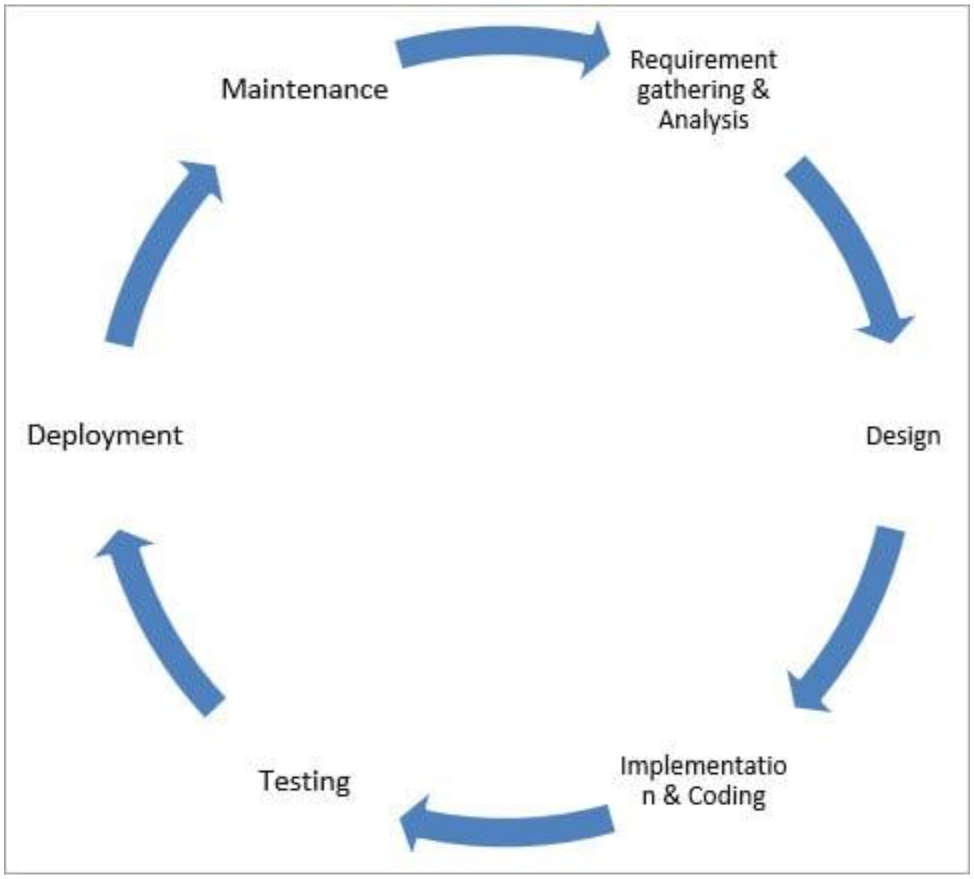


Figure 3.2: The Software Development Life Cycle (SDLC) [21]

model was chosen for this project. This is a variation of the waterfall model. The waterfall model can be defined as "a linear application development model that uses rigid phases: When one phase ends, the next begins. Steps occur in sequence, and, if unmodified, the model does not allow developers to go back to previous steps" [22].

Considering that this project was a foray into a field of computing science with which I was not yet experienced, I decided that versatility was of utmost importance for the development cycle. However, I also intended to split the development steps into clear and defined objectives which could be tackled singularly in an orderly manner. Therefore I decided upon the modified waterfall model, which allows for the developer to constrain themselves to a singular phase, but to also move between these phases freely. The modified waterfall model, defined as an extension of the waterfall model above, "allows a return to a previous phase for verification or validation, ideally confined to connecting steps" [23]. This can be better understood diagrammatically in Figure 3.3.

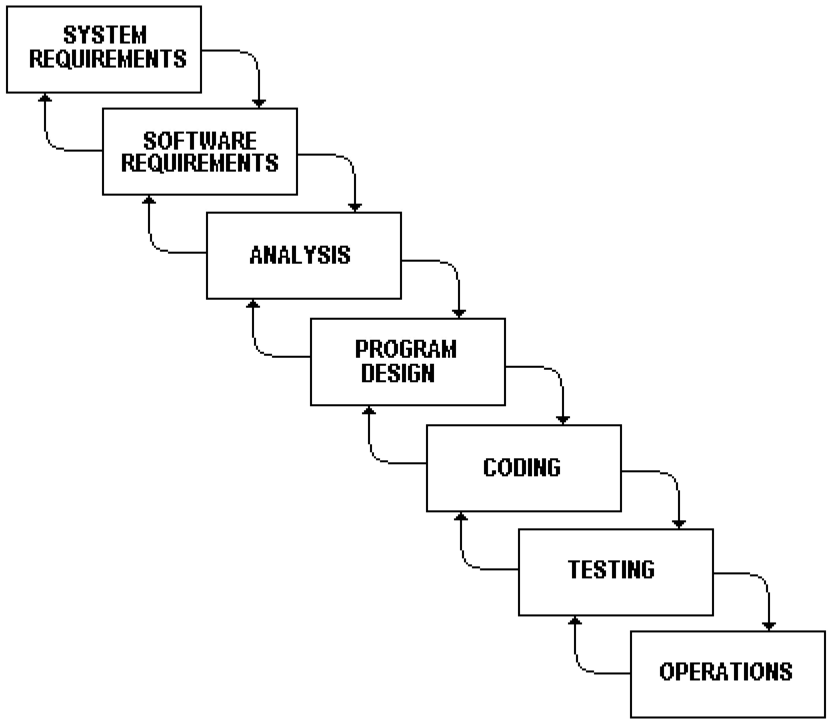


Figure 3.3: The Modified Waterfall Model [22]

The steps taken in the development cycle were as follows:

1. Documenting the system requirements based on project objectives
2. Using these requirements to decide on software to use in development
3. Synthesising the results of the two above steps to analyse how to best progress with the development of the program.
4. Designing the architecture of the program to implement the system requirements based on Step 3.
5. Developing the elements of the above architecture
6. Testing the individual elements of the program, and the simulation as a whole

As mentioned, the modified waterfall model allowed me to move between these steps, which allowed me to return to steps that needed further development fluidly.

3.4 Tools and Technologies Used

3.4.4 Unity

A game engine is “a software-development environment designed for people to build video games”. It’s a framework for developing that typically includes “a rendering engine ("renderer") for 2D or 3D graphics, a physics engine or collision detection (and collision response), sound, scripting, animation, artificial intelligence, networking, streaming, memory management, threading, localization support, scene graph, and may include video support for cinematics.” The purpose of these frameworks is “to enable simplified, rapid development of games in a data-driven manner” [25]. In essence, a game engine allows a developer to build their project without needing to program all of the elements of a framework that contribute to a functioning game. Creating a game engine for a modern game is typically an immense task, so many developers opt to use a prebuilt game engine like Unity.

Unity is a “cross-platform game engine developed by Unity Technologies” [24]. Unity profits through license fees – individual developers or small companies who generate less than $200,000 annually are able to use a free license. Those generating more than this amount in either revenue or funding must pay a monthly fee for each developer using the product [26]. Unity is one of the most popular game engines for public use, rivalled by Unreal Engine 4.

Unity was chosen for the development of this project because of its potential for seamless XR Integration [27], its free license model [26], robustness, and use of the C# programming language, which is explored further below [28].

3.4.5 C#

C# (C Sharp) is a “an elegant and type-safe object-oriented language that enables developers to build a variety of secure and robust applications” [29]. “It was developed around 2000 by Microsoft as part of its .NET initiative” [30]. C# is the chosen language of the Unity engine, and is the language that is used for any programming done by a developer using the engine, thus the programming for this project is all in C#.

3.4.6 Visual Studio

An IDE is “a software application that provides comprehensive facilities to computer programmers for software development” [31]. It allows developers to write code and then have that code be compiled and understood by the computer. An IDE is generally required for high-level programming and Visual Studio is recommended to be used with Unity due to Unity’s use of C#. Both C# and Visual Studio are developed by Microsoft [32].

3.4.4 Photoshop

Photoshop is a popular graphic design tool. I used it to create some art textures for the program.

3.5 Modelling

3.5.1 Unity Basics

Development of the project began with creating an environment within Unity that would match the objectives that were set out. The environment was to resemble the Urban Sciences Building.

The first step was to plan out how to replicate an existing building in the game engine. Rather than having 'free-form' control over building the environment, the project was restrained by real-life criteria that had to be represented accurately, and this had to be implemented using the tools that Unity provides. Unity has default assets used for creating objects in the game. For 3D games, this is several basic geometric shapes, including planes, quads, cubes, cylinders and more. There is also the option to 'import' prefabricated assets from external programs, such as Blender. Blender is a "a free and open-source 3D computer graphics software toolset" and can be used to create 3D art assets used in games, including animations and models [33].

The aim for the modelling aspect of the program, as highlighted in section 1.4, was to create an accurate model of the Urban Sciences Building, and this now was to be achieved using the modelling tools provided by Unity. To maintain accuracy, I used architectural resources online specific to the USB to recreate the floor plan, the dimensions, and the general 'look and feel' of the building [34,35,36,37].

Early prototypes were very basic, and proved that modelling such a large building realistically would be more time-consuming than originally anticipated in the development plan. An early version of the 3rd floor of the building can be seen in Figure 3.4

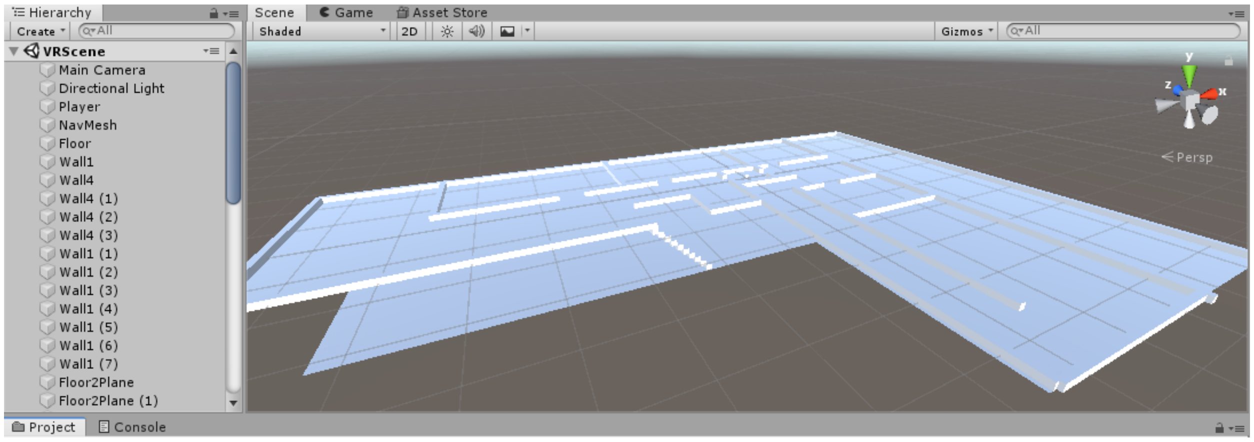


Figure 3.4: An early prototype

After developing a prototype, it became clear the number of different elements that would be involved in the modelling process to create a working environment in which to simulate the fire training process. These include but are not limited to:

- Complex geometrical shapes and structure
- Art assets (materials and textures)
- Lighting

- Navigational mesh
- Prefabricated objects
- Animations

I will detail the solutions found to these issues throughout section 3.5.

3.5.2 Probuilder

Using the default Unity tools to replicate the USB would have involved the use of many primitive shapes in order to recreate some of the more complicated shapes that exist in the building.

In order to tackle the issue of complex geometry, I used a recently added tool called Probuilder. This tool enables the creation of 'Poly Shapes' (short for Polygon). These were shapes that could be instantiated through creating several vertices anywhere on the X, Y, Z axes. This allowed for the complex structures that make up the USB to be more easily replicated than using the primitive shapes that are available typically in Unity.

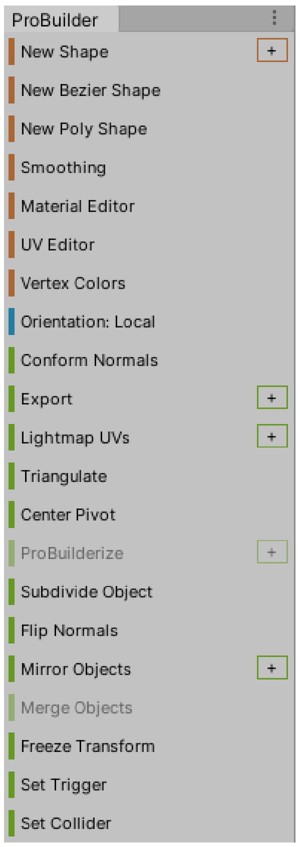


Figure 3.5: The Probuilder Menu

The other option to have solved this issue would have been to create assets in Blender and then import them into the project. After evaluating both choices, I decided upon using Probuilder. This was because:

1. Probuilder is integrated into the Unity engine as a plugin, and Blender is an entirely separate program. This meant shapes could be created efficiently and quickly.
2. Smaller and separate shapes would allow for more versatility when designing the levels, rather than larger, predefined objects imported from Blender
3. Probuilder allows for the editing of the vertices of 'Poly Shapes' after their instantiation
4. Using Probuilder would avoid issues with colliders for the object mesh and Navigation Meshes that could have been problematic when using Blender

I used Probuilder in conjunction with another tool called Progrids, which 'snaps' objects to a grid. This means that objects fit well with one another, with very little overlapping. Their values on the X, Y, Z axes are also more uniform integers, creating a more rigid, robust, and clean feel to the environment.

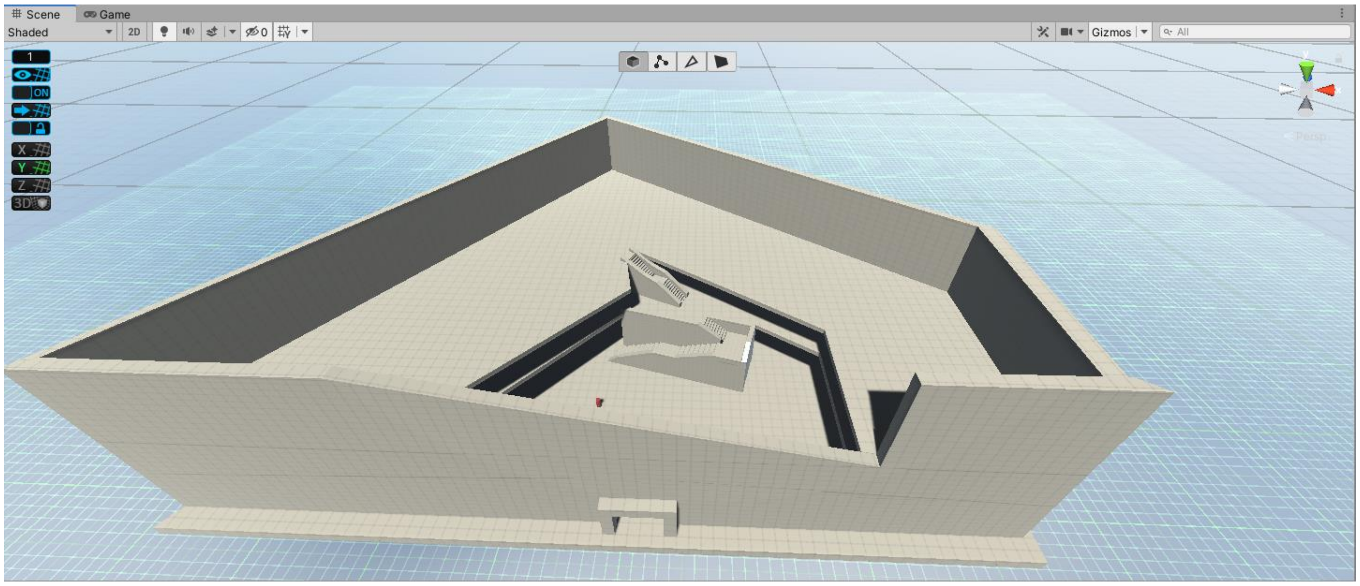


Figure 3.6: An early prototype using the Probuilder tool

3.5.3 Art Assets

One area that was hindered through this process was the art that could be applied to the shapes in the environment in the form of materials and textures. This affects the aforementioned ‘look and feel’ aspect of the design, that is important for immersion in a virtual reality experience.

Unity has an asset store from which it is possible to download assets, most of which are created by the community. The vast majority of these assets are paid. The free assets are generally poor in quality and were mostly irrelevant. Furthermore, it was not in the development plan nor relevant to the main aim to devote large parts of the schedule towards individually producing art assets to use in the environment. Therefore the final assets used in the creation of the building were some basic materials and textures, viewable in Figure 3.7.

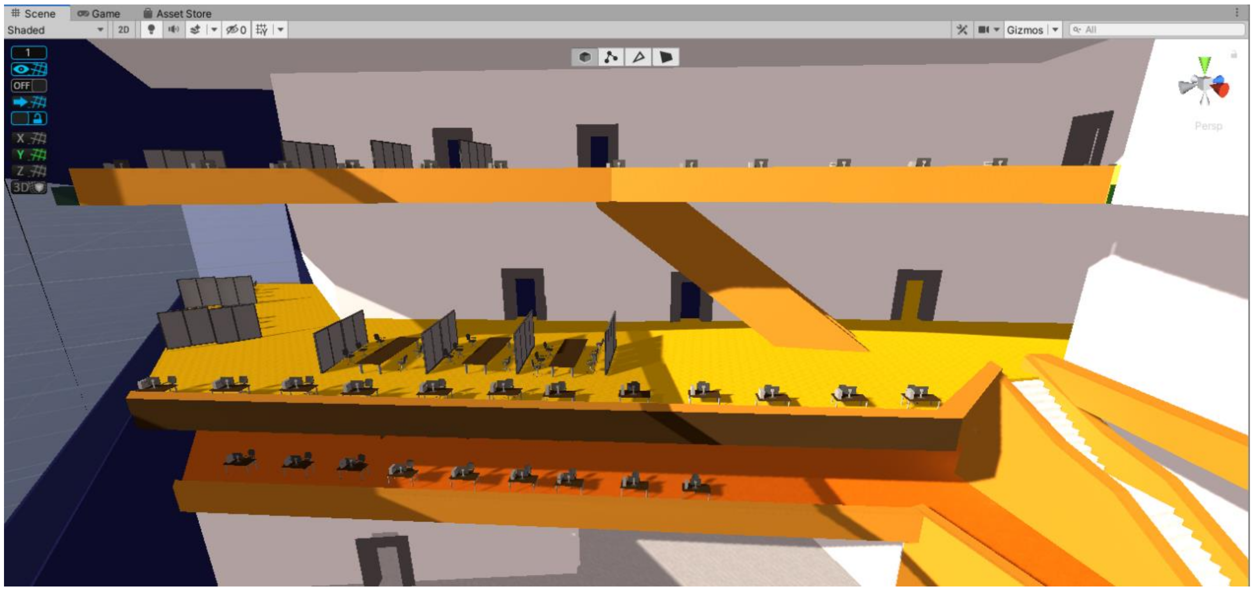


Figure 3.7: A view of the interior of the USB model

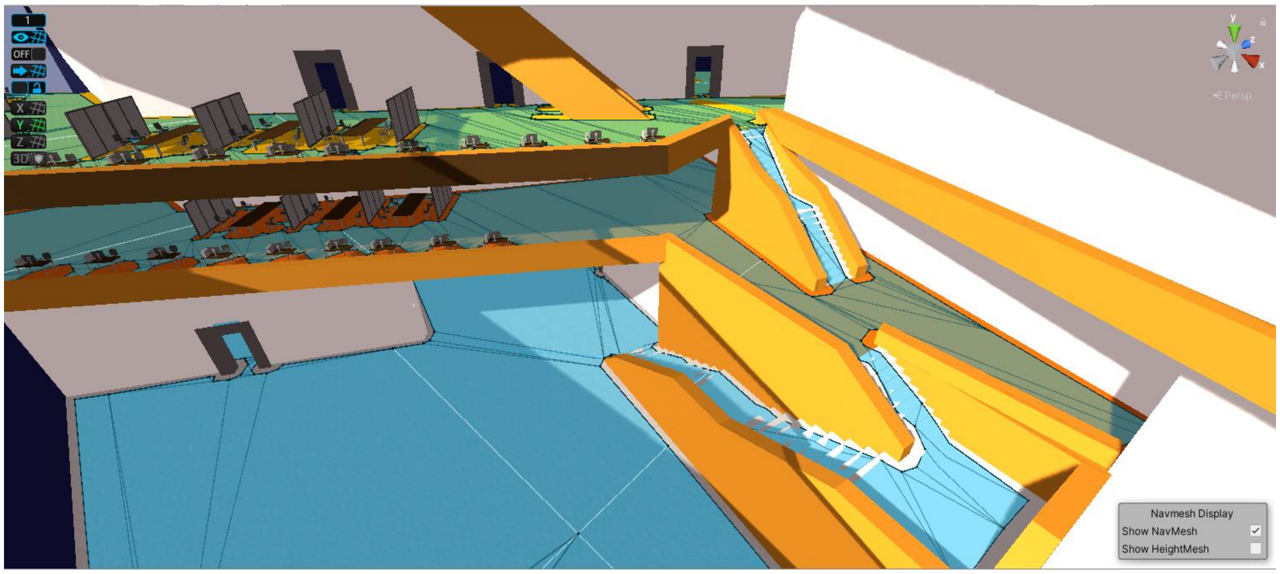
3.5.4 Lighting

The aim with the lighting of the environment was for a daylight scene. This was easy to implement using Unity's standard directional lighting, that mimics sunlight in a scene. Unity also offers a number of placeable lights, such as point lights, spot lights, and area lights, that have different impacts on the scene lighting. However, these lights did not fit well with the mood of the environment that was being created. Instead, I used Unity's ambient lighting feature to increase the light throughout the building. Furthermore, some of the many windows that exist in the real USB can be seen replicated in the model, allowing increased light from the directional light to fill the building. This was achieved by creating a material with a lowered alpha value and a transparent rendering setting to represent windows. This material was then applied to sub-faces of larger objects.

3.5.5 Navigational Mesh

A navigational mesh is part of the Unity engine's Artificial Intelligence (AI) system. It allows for the movement of AI Agents around the in-game environment. The Navigational Mesh defines which surfaces are traversable, and which are not. Elements of the environment that are 'Navigation Static' can be defined as 'walkable' which allows the engine to create a pathfinding map of the scene.

A Navigation Mesh was implemented for this project because AI was to be incorporated in the Evacuation scenario. It's possible to see the Navigational Mesh in Figure 3.8. The mesh is made up on the blue planes around the walkable areas of the map. Notice that non-walkable areas such as banisters, walls, and tables, are non-traversable by the mesh.



Show NavMesh ✓
Show HeightMesh

Figure 3.8: The Navigational Mesh

3.5.6 Prefabricated Objects

Once the structure of the Urban Science Building was complete, it was necessary to add some objects in the interior in order to have it resemble the real building. Using the primitive 3D objects provided by the engine would not have been sufficient in the level of detail~~,~~ and creating original art assets in an external program such as Blender was not in the scope of the development. Therefore, I chose to use free assets from the Unity asset store. Having found prefabricated objects for tables, chairs, and computers, I began to place these in the computer and office rooms within the building. This can be noted in Figure 3.9.

3.5.7 Animations

In order to instil a sense of realism in the game regarding the AI, it was important that there were realistic character models and animations. Using the Navigational Mesh, I had set up a system whereby some AI automatically evacuated the building, and some stayed inside, in preparation for the implementation of a training mission, the objective of which was to ensure all AI had left the building. At that time, all of the AI were represented by primitive cylinder shapes, which was detrimental to the immersion of the scenario.

Using the animator system, I created an animation flow control that allowed for character models to have an ‘idle’ and ‘walking’ animation. This animation control was accompanied by a script that determined whether the AI agent was moving and passed this information to the animator system. With this system in place, it was now possible to import character models available from outside of the engine, which allowed the AI to look like human characters.

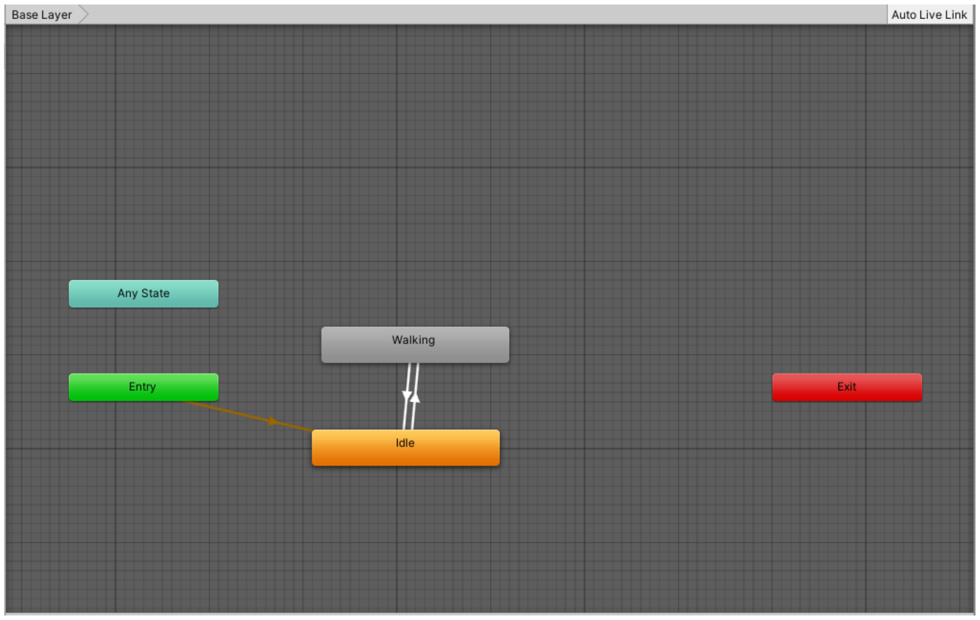


Figure 3.8: Animation flow control

The script involved was designed to test whether the AI Agent had reached its target destination or not. If the agent had not reached the target, then it would be in motion and thus the walking animation could be queued. With this system in place, it was now possible to have any number of AI agents simulate the evacuation of a building in the Unity scene. An example of this can be seen below in Figure 3.9.

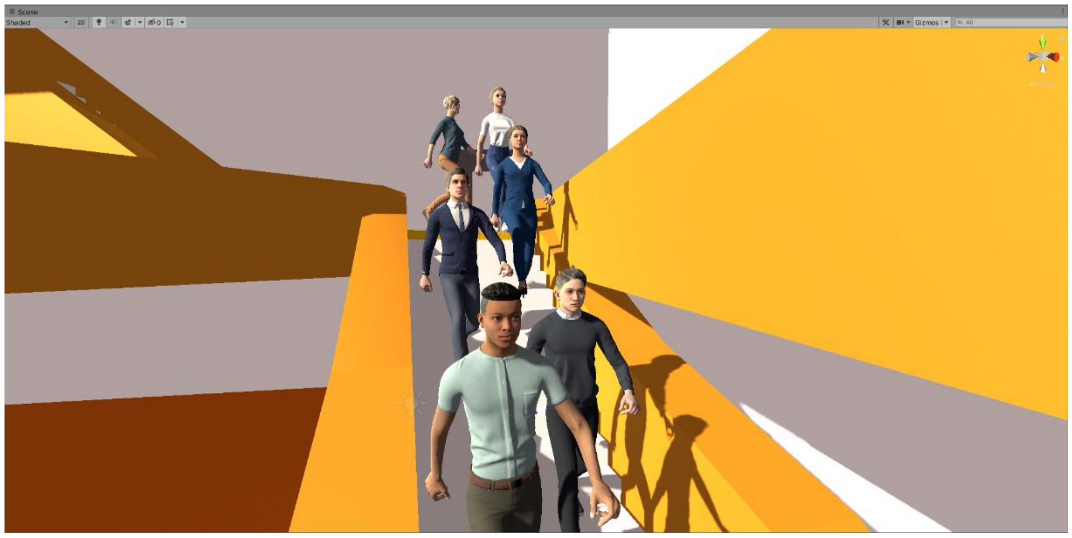
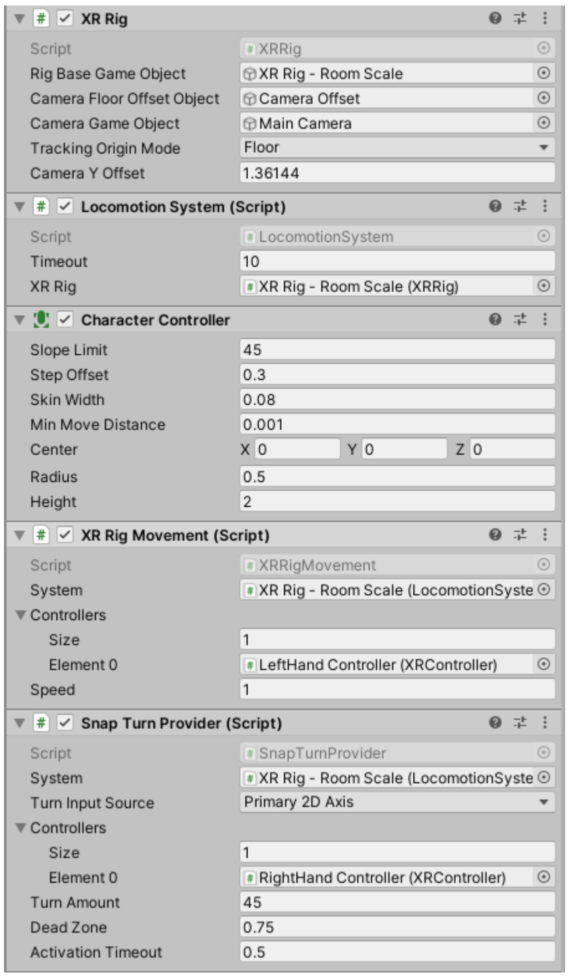


Figure 3.9: AI simulate the evacuation of the building

3.6 XR and Test Controllers

3.6.1 Unity XR Controller

The key objective for this part of the development was to ensure that the simulation would function with the use of VR. In order to do this, there needed to be a player controller that would accept input from both an HMD and hand-controllers, and this controller would need to be able to interact with the surrounding environment. The objective was to have a simulation that was compatible with a host of different VR devices, and so the controller would have to be adaptable to fit this specification. In order to meet this objective, I implemented a character controller using Unity's XR plugin framework. This controller was called the XR Rig, and was designed to accept input from any generic XR controller and HMD, allowing the user to simply connect their VR device and have it be compatible with the controller. This framework is referenced in Figure 2.4. The XR Rig is made up of a head object, a left hand object, and a right hand object. The hands have models which appear in-game when a VR device is connected. These models are there to mimic the player's hands.



XR Rig	
Script	XRRig
Rig Base Game Object	XR Rig - Room Scale
Camera Floor Offset Object	Camera Offset
Camera Game Object	Main Camera
Tracking Origin Mode	Floor
Camera Y Offset	1.36144

Locomotion System (Script)	
Script	LocomotionSystem
Timeout	10
XR Rig	XR Rig - Room Scale (XRRig)

Character Controller	
Slope Limit	45
Step Offset	0.3
Skin Width	0.08
Min Move Distance	0.001
Center	X 0 Y 0 Z 0
Radius	0.5
Height	2

XR Rig Movement (Script)	
Script	XRRigMovement
System	XR Rig - Room Scale (LocomotionSyste
Controllers	
Size	1
Element 0	LeftHand Controller (XRController)
Speed	1

Snap Turn Provider (Script)	
Script	SnapTurnProvider
System	XR Rig - Room Scale (LocomotionSyste
Turn Input Source	Primary 2D Axis
Controllers	
Size	1
Element 0	RightHand Controller (XRController)
Turn Amount	45
Dead Zone	0.75
Activation Timeout	0.5

Figure 3.10: XR Rig Script Setup

The XR Rig is room-scale, allowing the player to move around in their room and have this movement translated into the game. I also set up a static version of the XR Rig, but optimised the simulation for a room-scale version of the rig. I implemented a custom C# script to allow the player to use the hand controllers to move around the room. I opted for regular locomotion instead of teleportation. The reason that teleportation is the more popular option for movement in VR is because it minimises motion sickness. However, the teleportation aspect is less realistic than regular locomotion, and the players would not be in the simulation for long enough to cause motion sickness.

For the XR Rig to be able to interact with objects the simulation, it was necessary to use the XR Interaction toolkit to set up the controllers and interactable objects. The controllers used raycasting by default to interact with elements of the UI. Raycasting is the process by which 'rays' are cast out into the scene. In this instance, the controllers use raycasting to aim at objects and thus interact with them. By adding the XR Grab Interactable script to objects in the scene that could be picked up by the user, namely the fire extinguisher, it was possible to define which objects could and could not be 'grabbed' by the user.

3.6.2 Test Controller

Due to the lack of a VR device to be used for testing, for reasons mentioned in section 1.6, it was necessary to create a controller that was to be used for testing the game environment. I implemented a first-person controller, and used Unity's XR Framework to add a 'Mock HMD' setting to the game view, allowing me to essentially see what the game would look like through the screen of a real HMD.

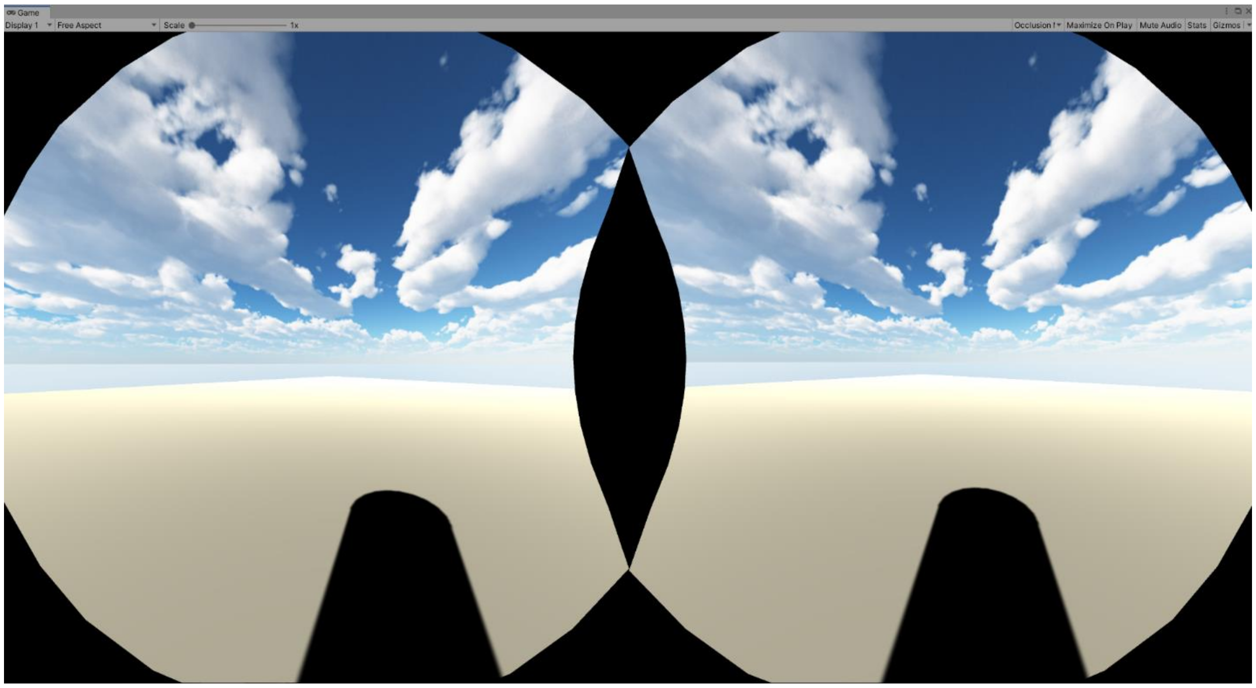


Figure 3.11: Mock HMD Occlusion in early development

Using C# scripting, I enabled this test controller to move around in the world, and look around using a mouse device. This test controller would then be equipped to interact with game objects in a parallel system to the XR Interaction toolkit.

3.6.3 Device Compatibility

Compatibility was a tough issue to tackle without the availability of a VR device. Through the use of plugins from the XR Framework, and the implementation of an XR Rig, it was possible to make the program compatible with major VR devices in theory. Using the test controller outlined above, I was able to interact in the game world in a similar way that the XR Rig would, which allowed me to make sure that most functionality was there. I also implemented a lot of functionality into the XR Rig, as discussed in 3.6.1.

The functionality for a 'plug-and-play' VR compatibility had been implemented in the project, however there was some limitation by the Unity game engine. Due to the XR framework being relatively new, there was not as much compatibility with the various VR devices as I had initially anticipated. Samsung Gear VR, Google Cardboard, and the

HTC Vive all used separate plugins or the deprecated VR framework, which meant that unfortunately I was not able to properly implement compatibility with these devices. The simulation would work best with an Oculus product, namely the Oculus Rift, as well as Windows Mixed Reality.

3.7 Training Implementation

3.7.1 Fire Extinguisher

Following the research from section 2.2.2, I chose to implement two different fire extinguishers in the game. The CO2 extinguisher would be used to put out an ‘electrical fire’ in one scene. In another scene, there would be a wood and paper fire, for which the foam extinguisher would need to be used. In both scenes, the user would need to use the information given to them in order to deduce which extinguisher to use. The different extinguishers can be distinguished by a universal colour scheme.

The fire extinguishers in the game were given a ‘foam’ particle system attached to the front of them. This foam could destroy ‘fire’ particle systems. The foam can be activated by player input. This is implemented as a key-press for keyboards, but could be activated by the use of the grip on a VR controller.

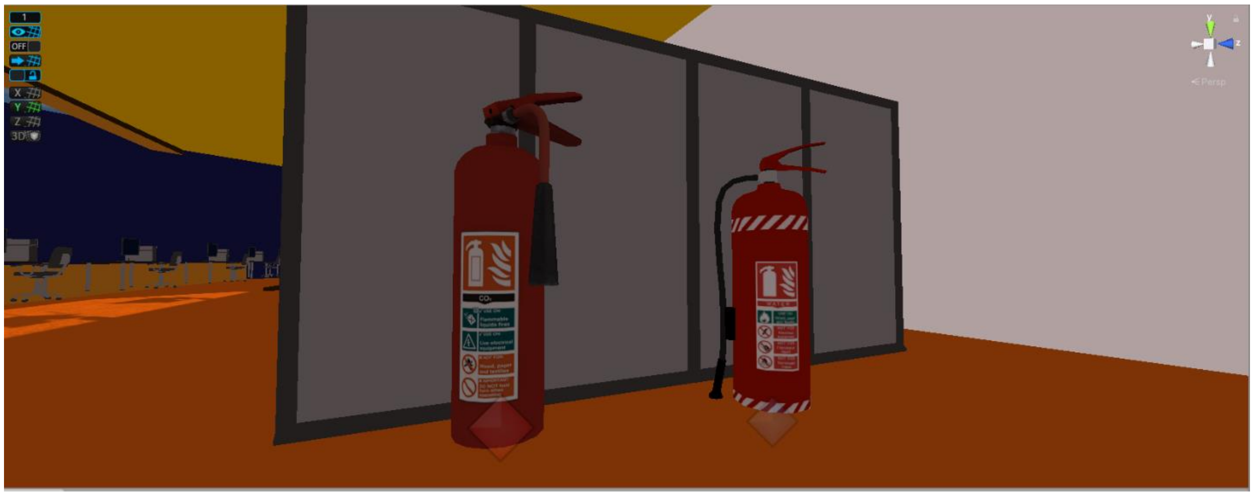


Figure 3.12: Fire Extinguishers

3.7.2 Search and Find AI

This training mission exists for members of staff working at the USB. The mission is designed for scenarios where a fire alarm is sounded, and yet some students may stay within the building, believing the fire alarm to be falsely activated, or a test. The scenario was designed to simulate checking the different rooms of the building, while a number of regular AI agents evacuate the building using a custom C# script. Certain AI agents are scripted differently, and stay in place until found and ‘triggered’ by the user.

Regular AI agents were scripted to leave the building immediately by the shortest path possible, and output the time taken. Some AI agents were given a script that meant that they would not leave the building until interacted with by the player, and would output the time taken upon reaching the exit. Using raycasting, the player would need to approach these AI characters, and use the designated keypress ‘F’, in order to interact with them, triggering part of their script which ordered them to evacuate the building.

3.7.3 Particle Systems

Through using Unity’s particle system I was able to implement both fire and foam systems. I created unique particle textures in Photoshop. Importing these into Unity allowed me to use these unique textures as the assets for my particle system.

Using a range of tools available through the particle system, I made the emission and colour of the custom particles imitate a fire. The outcome of this can be observed in Figure 3.13. By changing the size over lifetime, colour over lifetime, and several other effects, it was possible to create a simulated fire. This fire runs a ‘NavMesh Obstacle’ script, which means that the AI agents in the game must avoid it in their route to the exit of the building. At runtime, the fire dynamically ‘carves’ a hole in the Navigational Mesh. This carving is illustrated in Figure 3.14, where a gap in the blue surface that represents the mesh is visible.

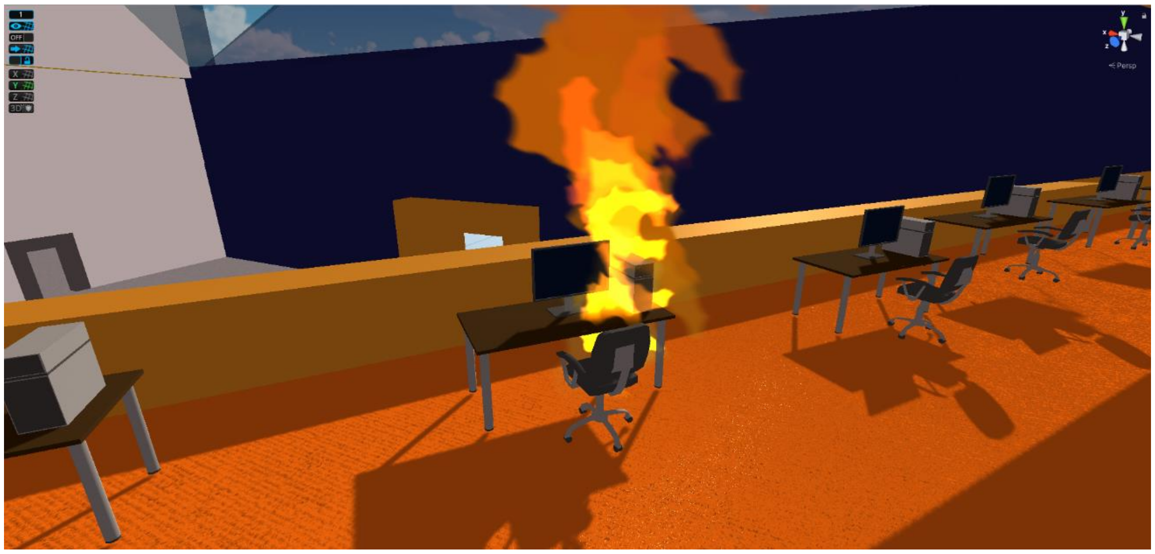
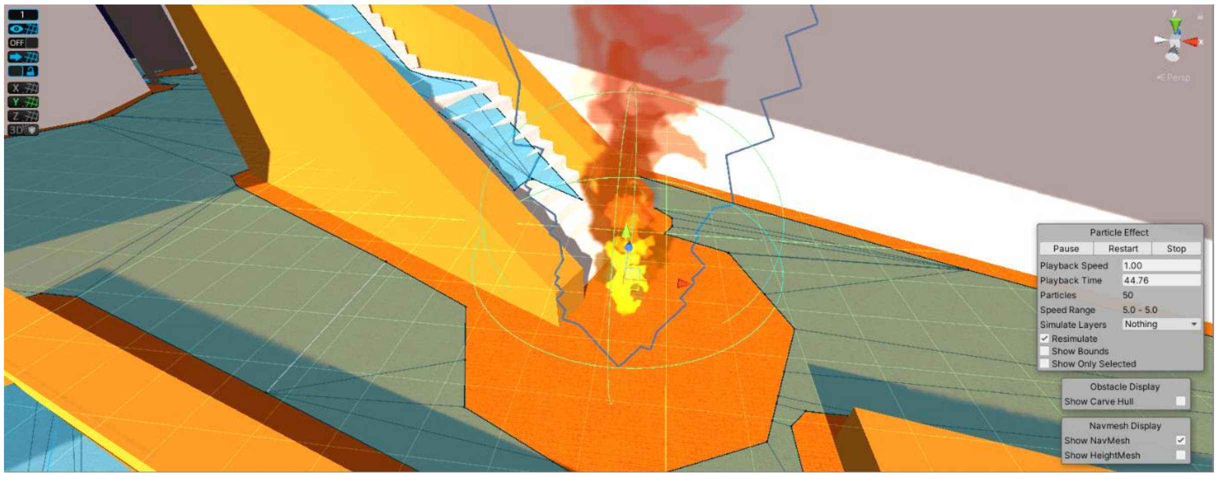


Figure 3.13: Fire Particle System


Particle Effect
Pause Restart Stop
Playback Speed 1.00
Playback Time 44.76
Particles 50
Speed Range 5.0 - 5.0
Simulate Layers Nothing
✓ Resimulate
Show Bounds
Show Only Selected

Obstacle Display
Show Carve Hull

Navmesh Display
Show NavMesh ✓
Show HeightMesh

Figure 3.14: A fire ‘carves’ the Navigational Mesh

The foam particle system is similar to the fire system, however the settings are altered to better represent the spray of foam over the spread of flames and smoke. This includes altering settings such as gravity, density, and alpha value over time. The foam particles are activated through the use of the fire extinguisher, and run a script to extinguish flame particle systems upon collision. The foam particle system is show in Figure 3.15.

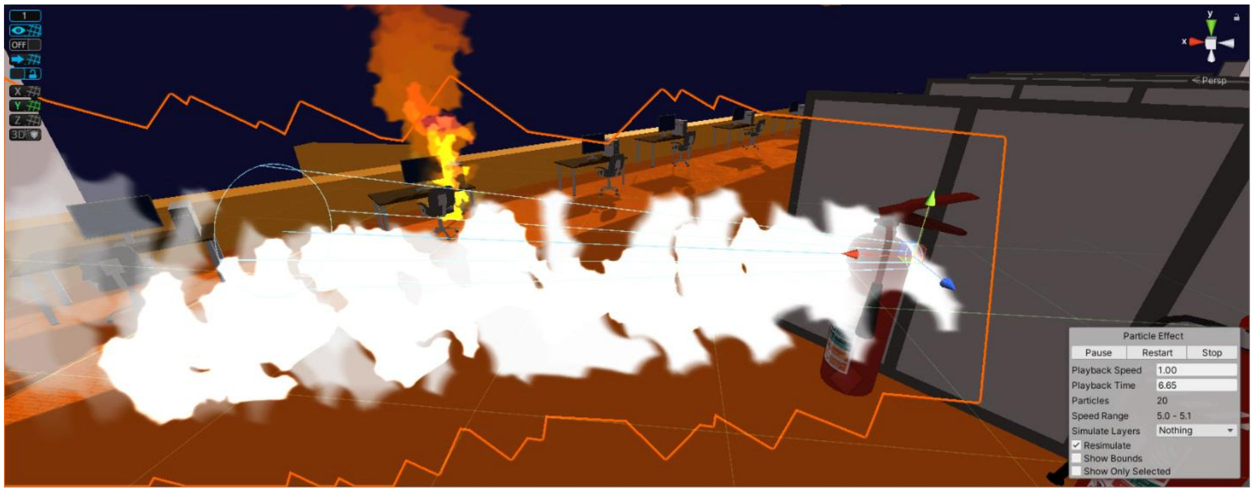

Particle Effect
Pause Restart Stop
Playback Speed 1.00
Playback Time 6.65
Particles 20
Speed Range 5.0 - 5.1
Simulate Layers Nothing
✓ Resimulate
Show Bounds
Show Only Selected

Figure 3.15: Foam Particle System

3.8 Summary

Chapter 3 has aimed to give detailed insight into the development process. The discussion of the software engineering aspects at the beginning allows the reader to understand the planning process behind the project, and how it adheres to the guidelines of software development, using the SDLC. After detailing the technologies that will be involved in planning this designed system, there is a high-level explanation on how the objectives of the project were met. This involves the process of creating a game environment in Unity, turning it into an XR compatible environment, and implementing fire safety training missions.

4 Testing, Evaluation and Results

4.1 Introduction

In this section I discuss the tests I performed on the project evaluate the efficiency and effectiveness of the simulation, as well as discuss the results. This section was limited by the inability to test with a VR device, or to test the simulation on a group of individuals, however I was still able to evaluate key elements of the project and test the simulation using a first-person controller.

4.2 Testing

4.2.1 Testing Strategy

With all of the functionality for the game implemented, it was important to properly test the different scenes in order to make sure that they worked. The main element to test was the training missions. It was important to ensure that the fire extinguishers worked properly, as well as the AI. Using the Test Controller that I implemented during development, I would rigorously test the different elements of the scene that together created the training missions.

4.2.2 Tests Performed

I performed tests on the model of the Urban Sciences Building. I was testing the colliders and navigability of the model. To do this I used the test controller to navigate to all areas of the building to ensure that all floors and geometry was properly placed, as well as layered. In order to make sure that gravity was applied properly, all walkable surfaces had to be under the ‘Ground’ layer. Without this layer, a negative y-axis velocity would accumulate while simulating gravity.

To test the AI, I placed agents in various locations around the USB, and made sure they were able to successfully navigate to the exit of the building. Furthermore, I ensured that results were output properly. Finally, I placed fires to block certain areas of the NavMesh to test the ability of the AI to avoid such obstacles.

In order to test the ‘search and find’ mission, I had to ensure that it was possible to ‘interact’ with the AI, and that the AI would move to their destination point after experiencing an interaction. To achieve this I repeatedly interacted with the AI at different areas within the building, and made sure that they evacuated properly following an interaction. To interact with an AI agent was to move within a raycasted range and use a keypress, designated to ‘F’. This would activate part of the script that ordered the agent to evacuate.

To test the ‘fire extinguisher’ missions, there were a number of elements to test. Firstly, it needed to be possible to pick up an extinguisher and point it at the fire particle system. Next, the extinguisher needed to ‘activate’ a foam particle system that would eject foam in a forward-facing direction. This activation would come through the input of a key-press.

4.3 Evaluation

4.3.1 Efficiency of the Simulation

Evaluating the efficiency of the simulation is to consider how different scenarios may affect the performance of the simulation. There are several parameters that can be changed within the simulation in order to achieve different outcomes and results. It is important to evaluate the performance of the simulation against different elements that can affect it, as a poor performance can lead to a low frame-rate, which can cause motion sickness as well as a very great detriment to realism in a virtual reality simulation. The optimal frame-rate in VR is 90 frames per second [36].

To begin with, the number of obstacles throughout the building can have an impact on performance. However, a lack of obstacles will take away from the realism of the virtual environment. Therefore it was important to reach a level of obstacle population that gave a good level of immersion without detracting from the performance. The Urban Sciences Building is a very large building, with many different objects inside. It would be difficult to recreate the same level of detail without having a large drain on performance. I hence opted to restrain the level of detail in order to keep performance optimal. I populated the building with desks, computers, and other prefabricated objects but did not include more detailed objects.

The number of AI agents would also tax the performance of the simulation. Again, the trade-off between performance and immersion had to be evaluated. Low performance would have an impact on the overall realism of the simulation. However, it was also important to have a high number of agents, both in order to simulate a real fire evacuation as well as to collect more data about the scenario. I found that 42 AI Agents offered a good level of detail without compromising the performance.

4.3.2 Effectiveness of the Simulation

Evaluating the effectiveness of the simulation involves understanding which permutation of the simulation offers the most safety and best achieves the main goal of an effective training simulator. The variables that would be modified to help evaluate the effectiveness would be the start location of the various groups of AI agents, the starting location of the fire system, the number of fires and extinguishers and the start location of the player.

To begin with, the placement of the AI agents combined with varying placements of the fire system allowed for the creation of a number of different scenarios, as the AI would take different paths to the exit based on these variables. These scenarios gave insight into the fire safety of the virtual model of the USB by showing the paths that the AI would take when certain paths were blocked by fire. The best placement to highlight this was fire particles blocking the 2nd floor stairs. This forced the AI from the 2nd and 3rd floor to use the lecture theatre to reach the 1st floor. This demonstrated the capacity for modelling the routes of people evacuating the building based on the location of the fire, without trapping the AI in the upper floors. However, it would be possible to make a scenario where the AI was trapped and the player must eliminate the fire using an extinguisher in order to allow the AI to escape. Essentially, the evaluation of these scenarios demonstrates the capacity of the simulation to learn about the outcomes of fire scenarios within the model created in the project.

The number of fires, extinguishers, and start location of the player would also determine the effectiveness of the simulation for training. I decided upon the use of two extinguishers for the player to choose between. This allowed for training users on the distinction between the types of extinguisher without overcomplicating the scenario. In the extinguisher scenarios, only one fire was set to spawn. This was because the focus of these scenarios was to identify the type of fire and to extinguish it properly. In the evacuation scenario, however, it was reasonable to set a number of fires in order to create varying evacuation routes. I therefore set two fires in the evacuation scenario to highlight this, however it would be possible to add more, in order to test blocking different evacuation paths within the model.

4.4 Results

Using AI I was able to generate results for the simulation. These results are the recordings of the time taken for AI agents to exit the building while avoiding fire obstacles.

AI Agent Name	Time Taken to Evacuate
Name: Character4(4)	Time: 7.366397 seconds
Name: Character5(5)	Time: 8.013126 seconds
Name: Character6(5)	Time: 8.272585 seconds
Name: Character4(5)	Time: 8.495649 seconds
Name: Character3(5)	Time: 8.789886 seconds
Name: Character1(5)	Time: 8.823299 seconds
Name: Character2(5)	Time: 9.142771 seconds

Name: Character6(3)	Time: 9.914858 seconds
Name: Character2(6)	Time: 10.01032 seconds
Name: Character6(6)	Time: 10.47344 seconds
Name: Character3(6)	Time: 10.9912 seconds
Name: Character5(6)	Time: 11.37083 seconds
Name: Character4(6)	Time: 12.1867 seconds
Name: Character1(6)	Time: 12.55087 seconds
Name: Character1(4)	Time: 16.97652 seconds
Name: Character6(4)	Time: 17.33175 seconds
Name: Character5(4)	Time: 17.81571 seconds

4.5 Analysis

These results show the potential for recording the routes of the AI as well as the time taken for them to evacuate the building in each scenario. This can be used to analyse how fire emergencies would impact the routes of people leaving the Urban Sciences Building in real life, based on the routes available to them.

The time taken for each agent to reach the destination point is dependent upon their placement in the building. In addition to this, the agents in the upper levels are hindered by the placement of a fire system on the stairs to the 2nd floor. This forces these agents to take a route through the Lecture Theatre. Hence the results of some agents are considerably longer.

4.6 Summary

To summarise, the different variables that make up the environment of the model can cause degradation in performance which must be evaluated in order to determine the efficiency of the project. Furthermore, the elements critical to the training aspect of the simulation reflect the effectiveness of the project, and I evaluated these to find the combination of these elements that would provide the most effective training for anyone using the simulation. Finally, the results gathered from the AI show how the placement of AI agents and fire systems allow for data to be gathered on routes of evacuation within the model.

5 Conclusions

5.1 A summary of the project

The project holds a well-modelled environment of the Urban Sciences Building. In this environment, the implementation of VR and first person character control mechanics allow the player to interact with AI and objects within the scene. The use of a fire extinguisher is implemented for both the VR controller and the test controller. The scenes are split into three missions which are designed to help the user train for the event of a fire scenario. Two of these missions involve choosing the correct fire extinguisher for a certain type of fire~~,~~ and using the extinguisher appropriately. The third mission is to find AI agents, representing students, who have not left the building after the alarm has sounded.

5.2 Satisfaction of Aims and Objectives

5.2.1 Satisfaction of the Main Aim

The overall aim of this project was to build models of Newcastle University buildings in a virtual reality environment, to introduce fire hazards and scenarios to these environments, and to develop training missions which could then be tested on a number of individuals. The data from these experiments could then be analysed in order to evaluate the effectiveness of VR training for individuals and to give insight into existing levels of fire safety.

The overall aim of the project was mostly implemented through development. Whilst handicapped by the lack of a VR Device to use for testing due to the COVID-19 university closure, I was able to use a parallel test controller with first-person mechanics in order to help implement and test most of the functionality I set out to create for the simulation. In conjunction with a rigorous adherence to the Unity XR framework, and the development of C# scripts to move, as well as the use of the XR Interaction Toolkit for the interactable objects, the simulation should work seamlessly with a VR Device.

To summarise, the main aim of the project was met to a good extent, with VR capability and properly designed training missions. The shortfall of the project was the testing stage, and this was, to a great extent, due to the University being shut down during Semester 2.

5.2.2 Satisfaction of the Objectives

This section will recap the original objectives and evaluate the extent to which they were satisfied by the project.

1. Create models of University buildings, specifically the Urban Sciences Building, that are playable levels and accurate to a good degree.

The model of the Urban Sciences Building is accurate. The structure and dimensions of the building are realistic, and will feel familiar to the player. The materials and textures in the model reflect those of the real building. However, the realism was limited by this particular area – individual developers are constrained by art when modelling in Unity. Paid assets are expensive, and creating unique assets requires artistic skill and is time consuming. Large studios and development companies generally have graphic designers to create art assets for their projects. Furthermore, with the limitations experienced by the project, it was neither necessary nor feasible to create further University buildings in the simulation. Overall, this goal has been sufficiently met, as the USB model is accurate to a good degree and allows for several playable training missions.

2. Design training that simulates several specific fire hazard scenarios in these environments, and clear objectives to solve them.

This objective has been completed successfully. There are 3 individual scenes in the Unity project, each with an individual mission. 2 of the scenes focus on the different types of fires and the appropriate extinguishers to use in the event of those fires, as well as the proper use of the extinguisher itself. The 3rd scene involves finding AI agents who have not left the building, simulating a scenario where some students may not leave the building, considering it a false alarm.

3. Implement VR compatibility in order to allow for this training to be performed with a Head Mounted Device and hand controllers.

Objective 3 has been completed to a satisfactory extent. During planning and development of the project, I ensured that I took the steps to implement the correct scripts and in-game objects/controllers to create a fully VR compatible experience. Handicapped by the lack of a VR device for testing, due to Coronavirus reasons as mentioned, I referred extensively to the Unity documentation to ensure that VR positioning, movement, and interaction were properly implemented and would work with simply attaching a VR-enabled device.

4. Combine the aspects of training, VR, and modelling within the game engine to create an immersive VR Fire Safety Training Program that can improve the reactions of users fire emergency scenarios

The different aspects that I set out to achieve welded together in the Unity engine well. What was created were functioning VR scenarios in a model of the USB. However, I think that the aspects of realism/immersion, and the extensiveness of the training program could have been expanded further.

5. Evaluate how data from the simulations can be applied to fire safety, and whether user reactions to fire scenarios are improved. In addition, evaluate the efficiency and effectiveness of the implementation.

Some aspects of this objective were not achieved to a satisfactory level, however overall it was completed satisfactorily. Unfortunately, I was not able to access a VR device during the later stages of development due to the shutdown of the University in the wake of the Coronavirus pandemic. As a result, I was not able to test the simulation on subjects in order to gather data for the project. I was able to generate some data using AI, but in the greater perspective this was not aligned with the objective I had set out to achieve. However, I did evaluate the efficiency and effectiveness of the project.

5.3 Personal Development

This dissertation was deeply challenging, as several areas of development were investigated in order to achieve the goal. To begin with, the project was an opportunity to use my learning from the past 3 years of my undergraduate degree in order to create a fully-fledged project of my own design. I was able to incorporate the software engineering and programming skills that have been taught during the course, and apply this to an area I had very limited experience in. With these skills, I was able to plan, design, and then implement programming as well as operate complex technological tools in order to create a working program that served a preconceived purpose.

The planning stage of this project reinforced the principles of the software development life cycle, academic research, and software engineering system design that had been instilled throughout my learning, and allowed me to practice these in the context of a serious academic project. Furthermore, it offered the opportunity to examine the commercial landscape of a growing technology industry, a skill which would be valuable for anyone developing a technology product.

The system development stage developed my existing technical skills and taught me many more. To begin with, I was able to explore an area of computing science with which I had great personal interest – virtual reality simulations. The opportunity to develop a functioning program in this area of computing was a privilege. In addition, I was able to master complex technological tools – particularly the overall use of the Unity engine. The project incorporates a wide variety of the elements that are found in almost every game, such as AI, lighting, UI, animations, and more. I became proficient with these tools during the development of my project.

Having chosen a Games Development dissertation project, I am confident that I am finishing the project with the knowledge as well as technical and research skills required to develop fully-fledged games, particularly the ‘serious’ games used for education and training that were the focus of this project. I was able to learn C# because it was required for the custom scripting of the Unity engine. Being able to learn and gain experience with a new language was a valuable addition to my technical skills, in addition to the aforementioned.

5.4 Future Work

There are a number of areas which I would like to develop further with this project. First of all, I would like to have access to an Oculus Rift, so that I can test the simulation properly, and gain real results from a testing process performed on subjects. This way, I could determine the effectiveness that VR training has over traditional methods of training, such as video teaching.

To perform this test, I would split users into two groups, and use the VR simulation to teach one group about the proper use of fire extinguishers and use a video teaching the same principles with another group. I would then evaluate the differences in recall and reactions between the two groups when quizzed on their knowledge, and upon mimicking the operation of a real extinguisher.

Further to this, the Oculus Rift would be useful for testing that the XR Rig controller works seamlessly with the interactable objects in the scene, which was an aspect of the simulation I was not able to fully test due to not being able to use a VR device.

The successful completion of objectives 1-4, as seen in section 5.2.2, built an interactive learning simulation that could be used as a foundation for a number of different simulations. The purpose for building a model of the USB (Objective 1) was to provide an interactive learning environment. Whilst there are currently several training scenarios implemented, the building as a whole has potential for further training and teaching, potentially not limited to just fire safety.

References

1. (2017) Virtual Reality Society. Available online at: https://www.vrs.org.uk/virtual-reality/what-is-virtual-reality.html

2. Milgram, P., Takemura, H., Utsumi, A., Kishino, F. (1994), Augmented reality: A class of displays on the reality-virtuality continuum. Proceedings of SPIE - The International Society for Optical Engineering. (Vol. 2351). Society of Photo-optical Instrumentation Engineers. Available online at: https://www.researchgate.net/figure/Simplified-representation-of-a-RV-Continuum_fig1_228537162

3. Williams, M. (2018). Checkmate Fire. Available online at: https://www.checkmatefire.com/workplace-fires-more-common-than-you-think/

4. Bannister, A. (2017). IFSEC Global. Available online at: https://www.ifsecglobal.com/fire-news/four-10-construction-workers-use-wrong-type-fire-extinguisher-electrical-fires/

5. (2017). Marsden Fire Safety. Available online at: https://www.marsden-fire-safety.co.uk/resources/fire-extinguishers

6. Unity Technologies (2020). Unity Documentation. Available online at: https://docs.unity3d.com/Manual/XRPluginArchitecture.html

7. Kinateder, M., Ronchi, E., Nilsson, D., Kobes, M., Müller, M., Pauli, P., & Mülberger, A. (2014). Virtual Reality for Fire Evacuation Research. In A. Krasuski, & G. Rein (Eds.), Federated Conference on Computer Science and Information Systems (Vol. 2, pp. 313-321). IEEE - Institute of Electrical and Electronics Engineers Inc. Available online at: https://portal.research.lu.se/portal/files/3483362/4610536.pdf

8. Lovreglio, R., Duan, X., Rahouti, A. et al. (2020). Comparing the effectiveness of fire extinguisher virtual reality and video training. Virtual Reality. Available online at: https://doi.org/10.1007/s10055-020-00447-5

9. Ably, T. et al. A* Search. Brilliant.Org. Available online at: https://brilliant.org/wiki/a-star-search/

10. Ably, T. et al. Djikstra’s Shortest Path Algorithm. Brilliant.Org. Available online at: https://brilliant.org/wiki/dijkstras-short-path-finder/

11. Oliva, D., Somerkoski, B., Tarkkanen, K., Lehto, A., Luimula, M. (2019). Virtual Reality as a Communication Tool for Fire Safety – Experiences from the VirPa project. Available online at: http://ceur-ws.org/Vol-2359/paper21.pdf

12. Z. Xua , X.Z. Lua , H. Guanb , C. Chena and A.Z. Rena (2014). A virtual reality based fire training simulator with smoke hazard assessment capacity. Advances in Engineering Software, Volume 68, p1-8. Available online at: https://www.sciencedirect.com/science/article/pii/S096599781300166X

13. Månsson, J. (2018). Using a virtual fire extinguisher as a tool for safety training. Available online at : http://lup.lub.lu.se/luur/download?func=downloadFile&recordOId=8936710&fileOId=8936711

14. Schlager, B. (2017). Building a Virtual Reality Fire Training with Unity and HTC Vive Available online at: https://www.vrvis.at/publications/pdfs/PB-VRVis-2017-008.pdf

15. Youtube (2017). Building a Virtual Reality Fire Training with Unity and HTC Vive.(Video) Available online at: https://www.youtube.com/watch?v=tgzOlAcyDNw

16. Statista. Available online at: https://www.statista.com/statistics/591181/global-augmented-virtual-reality-market-size/

17. (2020). Luminous Group. Available online at: https://www.luminousgroup.co.uk/project/premier-partnership-vr-fire-training/

18. (2019). Jasoren. Available online at: https://jasoren.com/portfolio/vr-emergency-training-simulator/

19. (2018). Huawei. Available online at: https://www-file.huawei.com/-/media/CORPORATE/PDF/ilab/education-training-ignite-vr-market-winwin-opportunity.pdf

20. Delaney, M. (2019). Survey: Education Among Top Industries for AR/VR Investments. Available online at: https://edtechmagazine.com/k12/article/2019/08/survey-education-among-top-industries-arvr-investments

21. (2020). Software Testing Help. Available online at: https://www.softwaretestinghelp.com/software-development-life-cycle-sdlc/

22. Conrad, E. (2011). Waterfall Model. Eleventh Hour CISSP. Chapter 8. Fig. 8.1. Available online at: https://www.sciencedirect.com/topics/computer-science/waterfall-model

23. Sridhar, B. (2015). Model Driven Software Engineering in the Mobile Era with an Emphasis on Security. International Journal of Electronics Communication and Computer Engineering. Vol 6. Issue 6. p 620. Fig. 1. Available online at:

https://www.researchgate.net/publication/284183672_Model_Driven_Software_Engineering_in_the_Mobile_Era_with_an_Emphasis_on_Security

24. Wikipedia. Unity (game engine). Available online at: https://en.wikipedia.org/wiki/Unity_(game_engine)

25. Wikipedia. Game engine. Available online at: https://en.wikipedia.org/wiki/Game_engine

26. (2020). Unity. Plans and Pricing. Available online at: https://store.unity.com/#plans-business

27. (2020). Unity. Manual XR Framework. Available online at: https://docs.unity3d.com/Manual/XR.html

28. Unity. Learning C Sharp for Beginners. Available online at: https://unity3d.com/learning-c-sharp-in-unity-for-beginners

29. Microsoft. Introduction to the C# language and the .NET Framework. Available online at: https://docs.microsoft.com/en-us/dotnet/csharp/getting-started/introduction-to-the-csharp-language-and-the-net-framework

30. Wikipedia. C Sharp. Available online at: https://en.wikipedia.org/wiki/C_Sharp_(programming_language)

31. Wikipedia. Integrated Development Environment. Available online at: https://en.wikipedia.org/wiki/Integrated_development_environment

32. Microsoft Visual Studio. Available online at: https://visualstudio.microsoft.com/

33. Wikipedia. Blender. Available online at: https://en.wikipedia.org/wiki/Blender_(software)

34. Hawkins Brown. Urban Sciences Building, Newcastle University. Available online at: https://www.hawkinsbrown.com/projects/urban-sciences-building-newcastle-university1

35. Hawkins Brown. The living laboratory, Urban Sciences Building Newcastle University Available online at: https://www.hawkinsbrown.com/cms/documents/Newcastle-USB_Digital-Case-Study.pdf

36. Newcastle Helix. Urban Sciences Building. Available online at: https://newcastlehelix.com/about/urban-sciences-building

37. Youtube. (2015). A walk through the Urban Sciences Building at Newcastle Science Central. Available online at: https://www.youtube.com/watch?v=pcKKhL1fjKk

38. Iris VR. The Importance of Frame Rates. Available online at: https://help.irisvr.com/hc/en-us/articles/215884547-The-Importance-of-Frame-Rates