



Dissertation

Blaine Burke

B.Sc.(Hons) in Software Development

MAY 9, 2021

Final Year Project

Advised by: Mr Gerard Harrison

Department of Computer Science and Applied Physics
Galway-Mayo Institute of Technology (GMIT)

Contents

1	Introduction	6
1.1	Objectives	8
1.1.1	Game Name	8
1.1.2	Game Logo	8
1.1.3	Art Styles	9
1.1.4	Screen distortion effect.	9
1.1.5	Target Audience	10
1.1.6	Player Movement	10
1.1.7	Player Interactions	10
1.1.8	Enemy AI	10
1.1.9	Player Weapon	10
1.1.10	Menu Area	10
1.1.11	Day and night cycle with fog effects	11
2	Development Diary	12
2.1	Features	12
2.1.1	Virtual Reality Rig	12
2.1.2	M1911 pistol	12
2.1.3	Enemies	13
2.1.4	Terrains	13
2.2	Change logs	14
2.2.1	Main Menu Area	14
2.2.2	Gun Range	15
2.2.3	Open World Area	16
2.2.4	Removed Features	18
2.2.5	telekinetic Grabbing	18
2.2.6	Vision Distortion	18
2.2.7	Mountain Climbing	19

<i>CONTENTS</i>	3
3 Methodology	20
3.1 Project approach	20
3.2 Testing	21
3.3 Version Control	21
3.4 Choosing technologies	21
4 Technology Review	23
4.1 XR Interaction Toolkit	23
4.2 Oculus Quest 2	24
4.3 Unity 3D	24
4.4 Trello Board	25
4.5 Google Drive	25
4.6 GitHub	25
4.7 C#	26
4.8 Visual Studio Code	26
5 System Design	27
5.1 Virtual Reality Controls	27
5.2 Enemy Path finding	28
6 System Evaluation	29
6.1 robustness	29
6.2 performance	30
6.3 Limitations	30
6.3.1 Graphics	30
6.3.2 Testing Space	31
6.3.3 Devices	31
6.3.4 Computers	31
7 Conclusion	32
A Appendices	35
A.1 GitHub	35
A.2 Installation	35
A.2.1 Build File	35
A.2.2 Unity Package	35

About this project

Abstract VR or Virtual reality is a relatively new, upcoming technology that had a market size of 10.3 billion dollars in 2019[1]. Today, according to statista.com, the virtual reality gaming revenue worldwide in 2021 is 1.4 billion dollars[2]. There are many large game companies that have created widely recognizable and popular virtual reality games. Companies such as Valve who created 'Half Life Alyx' , or Beat Games who created arguably the most popular virtual reality game 'Beat Saber'. I am using the unity engine to create my virtual reality game as it is free, and it is my preferred game engine for creating games.

Monkey Business is a virtual reality game built using Unity, a free 2D/3D engine and framework. It consists of multiple mini games such as zombie survival and rock climbing. In rock climbing you must climb as fast as you can to the end of the level in order to achieve the fastest time possible. If you successfully beat the current high score your score will display in the main menu area for that level! In zombie survival, the goal is to survive while also getting as many kills as possible! Upon dying the zombies drop additional magazines for the player to pick up and use. The high score is achieved by getting the most kills before losing all of your health.

Throughout this document I will be discussing the context of the game such as the reasoning behind a lot of my decisions. The Development Diary which is my personal documentation of the game development process such as small or large issues, finding assets and my change logs throughout the development of my project.

Authors My name is Blaine Burke. I am in final year of college doing a level 8 honours degree in software development in GMIT, Galway, Ireland. Since an early age I have always had an interest in video games. I watched hours upon hours of interviews with the creative minds behind my favourite games and watching videos of people playing them. Eventually I learned

that many games can be modded meaning that the game play, physics, or mechanics of the game could be altered in small or big ways. I saw the amazing things that people had managed to imagine and create and this inspired me more to want to create my own game. It wasn't until I began college that I really began making games. I had a module called 'Mobile Application Development' with a lecturer named Damien Costello. He taught me all the fundamentals of Unity and C#. It was then that I finally began my passion of making games.

Chapter 1

Introduction

VR or virtual reality is a virtual experience that can be designed to be similar or different to the real world. It is a relatively new, upcoming technology with a huge future potential and in 2019, had a large market size of 10.3 billion dollars [1]. It has vast amounts of uses for research, development, and training, throughout industries such as medicine, engineering aviation and rehabilitation. It has many obscured uses such as helping people overcome a physiological phobia or social marketing which involves using commercial marketing principles and techniques to enhance the lives and welfare of people.

A very important aspect of virtual reality is immersion. Virtual reality is the term used to describe a three-dimensional, computer-generated environment that can be explored and interacted with by a person. That person becomes part of this virtual world and is immersed within this environment. Whilst there, they are able to manipulate objects or perform a series of actions. If that person does not feel immersed the experience will be heavily affected. The feeling of immersion is captured due to your perception of things, how you can hear, see or become aware of things through your senses.

In an article written by Rebecca A. Penn and Michael C. Hout of the New Mexico State University named 'How VR "Tricks" Your Brain'[3], they wrote

"The brain's interpretation of the senses which create our understanding is called perception. The process of our brain interpreting our senses into experiences."

For example, imagine a ball bouncing in front of you. You can see a ball bouncing, hear it bounce and if it hits you, you can almost feel it, and it's all these things that through our perception are combined creating and allowing

us to experience it.

New technologies to improve virtual reality immersion are also being released constantly such as haptic feedback vests. They work by having multiple feedback points that will vibrate depending on how hard you've been hit and where you have been hit on your virtual body. This allows you to actually feel as if you have been, for example hit by a ball. The ball would impact your virtual body and you would feel it through the haptic feedback vibrations. Another new technology to greatly increase immersion is the KAT WALK C. It is a circular treadmill that the user can walk on to simulate walking inside the virtual world. It allows the user to feel as if they are walking around in the real world. With these new technologies, virtual reality's immersion has been greatly increased and in the coming years due to inventions like these, virtual realities immersion will be constantly increasing.

Monkey Business is a family friendly game created using Unity 3D. It consists of only two mini games currently, rock climbing and zombie survival. The aim of rock climbing is to climb as fast as possible to end the end of the level. At the end of the level there is a red button which the player must hit in order to end the timer. If they beat the current high score for the level the timer's text will appear green and if they do not it will appear red. All of the high scores for the levels will be displayed in the main menu area on an interface. In zombie survival the player must fight hordes of zombies in an attempt to survive as long as possible whilst getting as many kills as possible. The high score is achieved by getting the most kills.

When I initially began this project, I had no virtual reality hardware. I spent a week researching different options and eventually narrowed my options down to two, the Oculus quest 2 or the Valve Index. They are both excellent headsets but there were several differences between them. The Oculus Quest 2 can be used in both a wireless and wired mode, it also has hand tracking and a higher resolution display. The Valve Index on the other hand also has many benefits such as individual finger tracking, a wider field of view and a higher refresh rate. However, there are significant differences in the prices. The valve index costs over one thousand euros whereas the Oculus quest 2 is priced at 349 euro for the 64-gigabyte version and 450 euro for the 256-gigabyte version. Due to being a student and having not too much disposable income, I chose the latter.

The files used in this project can be found in my GitHub repository.
[Click here to view it!](#)

1.1 Objectives

I began by setting out some objectives for my project such as the theme, the game play mechanics, the target audience et cetera. Throughout the course of this project, many of these objectives changed. Some were for technical reasons and others were due to my imagining of the end result changing. My original objectives were as follows...

1.1.1 Game Name

Originally, I planned on calling my virtual reality game 'Xylophobia', meaning a fear of woods especially in the dark. I took inspiration for the name Xylophobia from the game Phasmophobia which means an intense fear of ghosts. This name was fitting as originally the game was an open world game that takes place in a forest where the player could freely roam around the area. There was a day and night cycle in the game, and upon the game becoming dark the player must stay in range of a light source in order to avoid their screen becoming distorted with a visual effect making it difficult for them to see. This distortion represents the effects a phobia can have on someone, in this case Xylophobia.

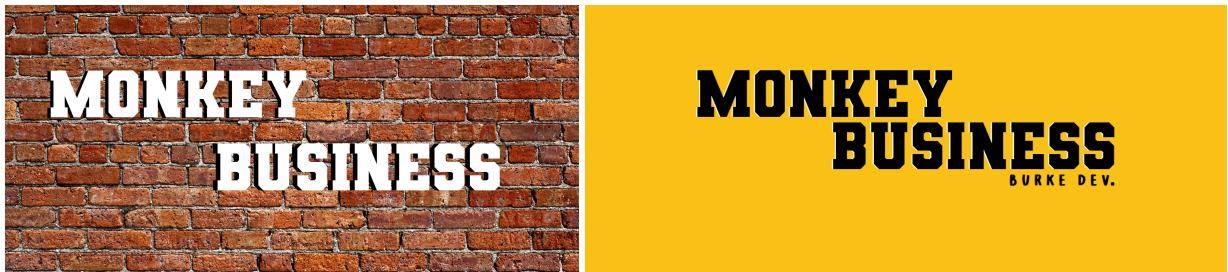
Unfortunately, due to the game being too CPU intensive, I needed to change to a stylized, low poly art style. The assets I chose were bright, colourful and family friendly which were not fitting for a horror game. Therefore, I decided to give the game a more suiting name, 'Monkey Business'.

1.1.2 Game Logo

When creating the logo, I wanted it to stand out. I wanted to do this by using contrasting colours. I researched logos for companies such as Fanta and IKEA which both have their logo or company name in a contrasting colour to the background, for example IKEA's yellow on blue. This method makes the logo stand out and eye catching. I decided I would make a few different concepts for the logo.

My first concept was to put white bold text on a brick background. I put a black shadow effect of the writing to make it stand out even further. My second concept was to keep a simple design with solid colours that, again contrast each other. I chose to use a yellow background and black bold text with a white shadow effect on the text. I chose black and yellow as they are colourful, and it gives the best visual contrasts. I asked family, friends and

fellow college students' which logo they preferred and the latter option was chosen.



1.1.3 Art Styles

Upon beginning the project I initially used realistic detailed art styles. This was to give a grim, realism to the game to emphasise the dark nature of a post-apocalyptic world. Unfortunately, I encountered many issues with my original ideas. The realistic art styles proved to be extremely CPU intensive. I attempted to fix these issues by using LOD groups, occlusion culling and other rendering techniques. Sadly, they weren't enough to fix the frame rate issues. As a result, I began to re-imagine the game and its art style. After researching many virtual reality game art styles for many virtual reality games such as Phasmophobia and Superhot VR. Both these games have simple low poly graphics and because of this they are easier to run, improving accessibility for lower end systems. I some assets in the asset store that were low poly yet detailed, created by Broken Vector [4]. Using these assets, I created several aspects of my game such as the main menu and the zombie survival area.

1.1.4 Screen distortion effect.

There is a day and night cycle in the game, and upon the game becoming dark the player must stay in range of a light source in order to avoid their screen becoming distorted with a visual effect making seeing objects from a middle to far distance very difficult to see. This effect is my attempt to display how hindering a phobia can be to a person.

1.1.5 Target Audience

The target audience of the game was originally meant for horror fans and an older age group. This is due to the game being a horror game featuring zombies, weaponry and blood effects. Upon changing the game to Monkey Business however, I removed the blood particle effects and chose new assets due to the game being a more family friendly game.

1.1.6 Player Movement

Implement a continuous movement system so the player can freely move around the play area using the left analog stick. This also includes snap turning so the player can rotate 30 degrees to the left or right using the right analog stick.

1.1.7 Player Interactions

I wish to allow the player to grab and interact with specified objects. This includes the pistol and climbing walls.

1.1.8 Enemy AI

Originally, I planned to have the zombies search for the player and if the player became visible the zombies would run at them and attack. I removed the searching aspect after I changed the game from Xylophobia to Monkey Business and changed it to just path finding. This was done in order to give it a more arcade game feel.

1.1.9 Player Weapon

To add a working pistol that can be loaded and cocked for the player to use in order to defend themselves from zombies.

1.1.10 Menu Area

Create a menu area that has a main menu, high score interface and some things for the player to play around with. Originally I had created a prison area that had a climbing wall and a gun range, where the player could get a pistol and practice shooting some cans. This was done in order to act as

a tutorial and allow the player to learn how to play the game at their own pace. The gun range area was removed upon the change to Monkey Business and currently there is a climbing wall and monkey bars for the player to use.

1.1.11 Day and night cycle with fog effects

Create a day and night cycle for the game as it makes the game world look more beautiful and allows me to implement the screen distortion effects. I added fog as it increases fps in a game. This is as it lessens the amount of objects and textures to be rendered on screen so it is less taxing on the players hardware and makes the game easier to run.

Chapter 2

Development Diary

Throughout the course of the project I recorded my changes and additions to the project through GitHub. However, due to my projects sheer size I could not upload everything directly to GitHub. Instead I recorded all main updates and changes in the GameUpdates.md file and uploaded my game build and package to google drive.

2.1 Features

2.1.1 Virtual Reality Rig

When I began the project, the first thing I created was the virtual reality rig. Using YouTube tutorials by Valem[5] and guides on stack overflow, I eventually created the head of the virtual reality rig. I then began to create the virtual reality hands for the player. I began by finding a hand model with the fingers all being child objects of the hand. This allowed me to add animations to the hands such as a fist and pinching hand gesture. I also created it, so it takes into account how much the buttons have been pressed by getting the input devices float values. The hand animations will work in sync with the amount the button has been pressed. For example, by pressing down the grip button to the half way position, the hand gesture will not be a fully formed fist but will be a half formed fist gesture.

2.1.2 M1911 pistol

When creating the pistol, I had many issues. This is due to it having many moving components such as the pistol slider and the magazine. I began by finding a pistol asset that was split into several parts. This allowed me to add

joint components onto the individual moving game objects and add scripts to allow the player to use the gun. I began first by attaching the XR Grab Intractable script to allow the player to be able to grab and hold the gun. I then ran into an issue where the pistol was not being held correctly in the players hand, so I created a hold transform for the pistol. I then added a sphere attach point for the pistols magazine that allowed the player to insert a magazine into the pistol. Once again, I had to create an attach transform so the magazine would sit in the correct position. I had a lot of issues when creating the slider for the pistol. The slider would detach from the gun when grabbed but once released would reattach itself. Eventually I overcame this issue and created my functional weapon.

2.1.3 Enemies

The enemies were originally meant to patrol an area and until the player entered within a radius of them. This method was functional to a degree, however there was no path finding on the enemy so it would just walk in a straight line toward the player. I then began to research path finding algorithms such as Dijkstra's algorithm or the A* algorithm. I eventually chose to use the built-in path finding system in unity called a NavMeshAgent. It is a component that is attached to a mobile character in the game to allow it to navigate the Scene using the selected NavMesh. It is a simple yet effective path finding method. The enemies could now search for the player and chase them successfully using path finding. I then searched on the unity asset store and I found many different assets that I could use for my game. I also used other sites such as Turbo Squid to search for assets [6]. I eventually chose a simple, cartoon zombie model for my game that came with its own animations for use.

2.1.4 Terrains

When beginning this project, I intended to have the play area as an open world, rural mountainous region. To do this I used Unity3D's Terrain Editor. It is a very useful, powerful tool that allows you to quickly create hills, mountains, plateaus, valleys, and more. I used it to create a mountain in the center of the area, some hills, roads, paths and forests. I created many different versions of the open world area which unfortunately were all removed from the game.

2.2 Change logs

Throughout the course of this project, many features and ideas were removed. They were removed for many reasons for example being too intensive to run or not being suitable toward the game, its theme and its art style.

2.2.1 Main Menu Area

The main menu area has vastly changed since its first build. Initially it was a prison area where the player could also freely roam around and learn the mechanics of the game for example by using the climbing wall. There was a vast amount of graphical changes too. The picture bellow shows the vast amount differences between the main menu area in version 1.0 and version 1.1 of the game. There was many things changed between them such as ...

1. The addition of a day and night cycle to increase immersion in the game.
2. Ambient sounds to increase immersion into the game world.
3. Fog effects that change depending on the time of day to make the world more realistic and beautiful.
4. New textures for the ground.
5. The addition of a shooting range area for the player.

Ultimately, the prison area menu was removed from the game after version 1.3 was released as it was not suitable for a family friendly game. The menu was changed for a bright, colourful park area with a small playground featuring a climbing wall and monkey bars for the player to play with as can be seen in the image below. I also created two floating menus that constantly rotate to face the player at all times. These menus contain the the main menu and the high scores.



2.2.2 Gun Range

During versions 1.1 and 1.2, there was a gun range in the main menu area. Here the player could find a pistol and some magazines that they could use to shoot targets. In version 1.1, the targets were some zombies that would freely walk about outside of the prison area. In version 1.2, the zombies were replaced with some cans that the player could shoot and knock over. This

allowed the player to learn how to use the weapon before playing the game itself by acting as a tutorial.



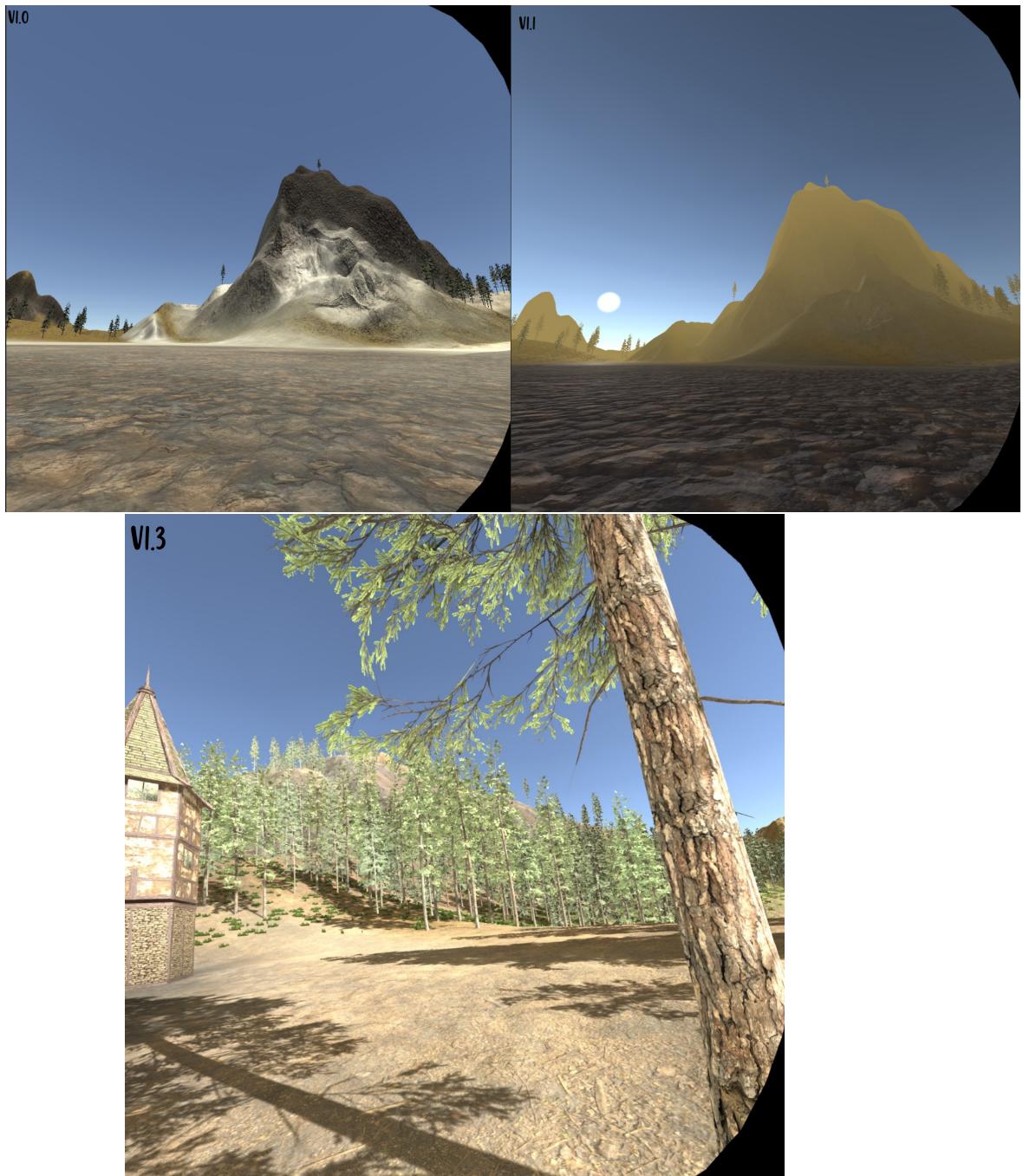
Due to being a part of the prison area, it was removed also during update 1.3. Currently there is no gun range in the main menu area as it would not fit the theme of the park area. I will replace it at a later date with a tutorial area where the player will be instructed on how to use the gun.

2.2.3 Open World Area

From versions 1.0 to 1.2, there was a large open world area that was originally, going to be the zombie play area for the player. This area was a mountainous region that contains a large forest, small villages and many pathways. Before I added grass, fog effects, shaders and a day and night cycle the world looked quite bland as can be seen below on the first image. After adding some trees, fog and the day cycle, the area provided the player with beautiful views as can be seen in the second image. The ambient sounds of birds singing in the forest could also be heard. This was to add life to the forest and to further immerse the player into the game world.

After update version 1.3, more details were added to the world further enhancing the beautiful scenery. I did more detailing on the textures of the mountain and paths by mixing gravel and dirt layers together to give a dirt path type effect. I also added the ability for the trees to sway in the wind and improved their placement positions. I added many different colours of

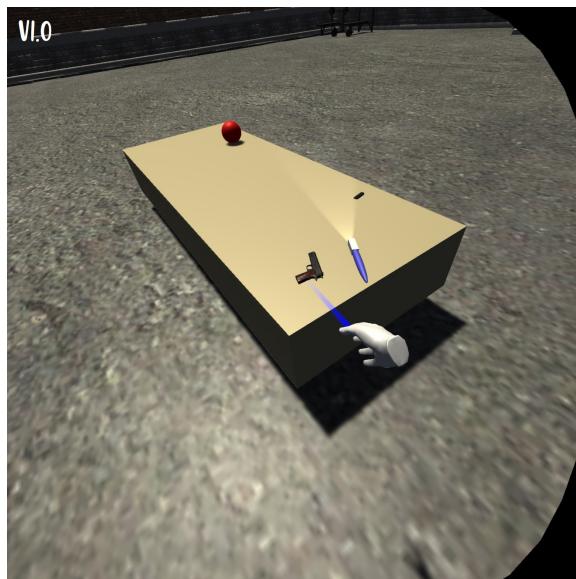
grass and flowers to make the terrains aesthetics less bland and improve the immersion of the game. All of these aspects combined made the game world beautiful and realistic as can be seen in the third image below.



2.2.4 Removed Features

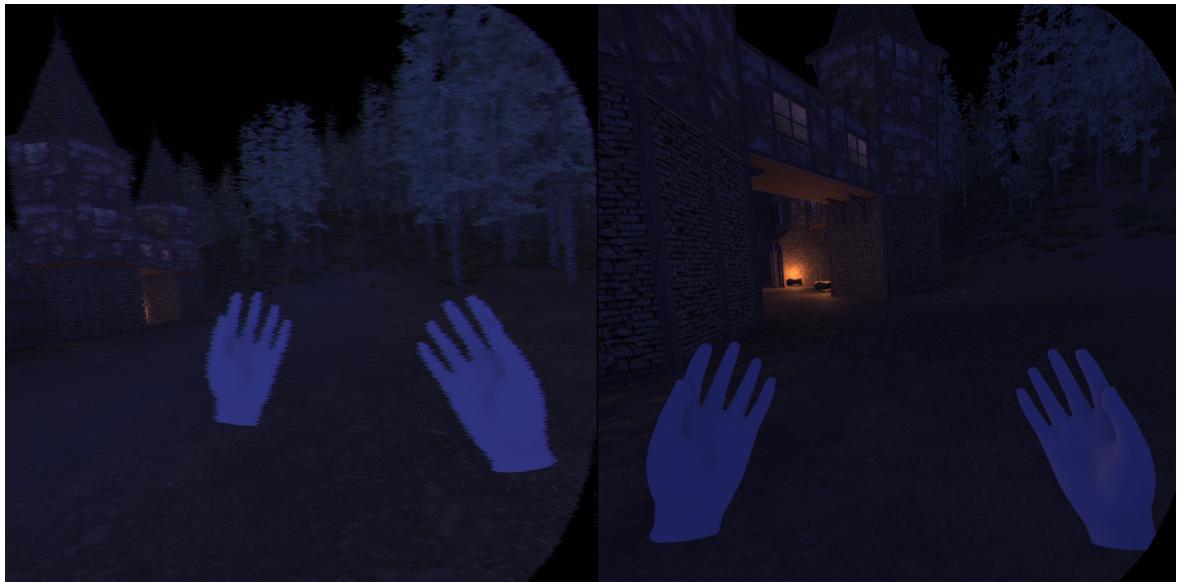
2.2.5 telekinetic Grabbing

During update V1.1, the player had the ability to use telekinesis to grab items. To do so the player would point their hand towards the desired item and a blue laser would appear from their hand. Then they would press the grab button and the item would get pulled into their hand. It was a very cool and useful feature but was removed during update 1.2. This was due to the game being realistic, so I wanted the player to have to pick up items themselves. In the future of the game, I could potentially find a use for it again.



2.2.6 Vision Distortion

Xylophobia means a fear of the forest especially in the dark. Due to the game originally taking place in a zombie infested forest and featuring a day and night cycle. Once night falls, if the player is not in range of a light source, their screen would become distorted as seen below. There were different types of light sources, small, medium and large ones and each one had a different radius of effect. Being within this radius would stop the distortion effects and allow the player to see clearly again. This effect made it harder for the player to see in the dark in an attempt to simulate how much a phobia can drastically effect someone. However upon changing the game theme and name, this feature became redundant.



2.2.7 Mountain Climbing

The climbing aspect in the game was not always rock climbing but was first mountain climbing. I had created multiple stages where the player would have to climb up large cliff faces to reach the end. There was no timers or high scores to beat. The player had a head light attached to them in order to allow them to see in the dark. The downside of these climbing levels is the sheer size of them. The levels would take approximately 4 to 5 minutes to complete. Upon switching to monkey business I removed the long mountain climbs and changed to rock climbing instead.

Chapter 3

Methodology

3.1 Project approach

Due to my projects large size, I took an iterative approach when doing this project meaning I created all the required game objects and scripts and then came back to them one by one, fixing and improving them. I enjoyed this approach as I was doing different things every two to three days, which kept it interesting for me and allowed me to further enjoy the development process.

Initially I began by creating the basic game objects for the game such as...

- A VR rig with simple movement scripts for the body, hands and head.
- A basic test level for development of other future game objects.
- Enabling the player to grab and hold specified objects.
- A basic enemy AI that simply walked directly toward the player when within its line of sight.

They were all functional but needed improvements for example the enemy would always face the player but sometimes would get stuck on objects and would not be able to move toward the player. I began dealing with any issues with my basic requirements by either fixing or removing things from it. For example, I eventually removed my first enemy as it was quite buggy and not very intelligent as it would get stuck on objects due to having no path finding algorithm implemented. I continued this cycle throughout the entirety of my project as it proved successful and efficient to me, being a solo developer.

3.2 Testing

I took a trial and error approach to testing. I would slightly tweak things to see how well they would work and what affect it would have on its functionality and surroundings. For example, when creating grab positions for the player, I would have to tweak the transform of the held item so it is positioned correctly in the players hand. I found this time consuming as I would have to stand away from my desk, put on the virtual reality headset and remotes and test a feature for sometimes only a few seconds..

I was also affected by being a solo developer as I would sometimes not notice a small mistake either in Unity or in the scripts that someone else might notice. These small errors in my code initially were frequent however over the course of the project, due to improving my knowledge of Unity and C#, I would rarely make simple mistakes. It also effects the amount of time I can spend on certain aspects of the project such as the enemies or the level design.

3.3 Version Control

I used GitHub to upload my files but quickly ran into an issue due to the sheer size of my project. The overall size was almost ten gigabytes due to assets and other large files. To combat this issue, I only uploaded scripts, audio files and other small files to my repository. In the README.md file, I then provided links to my google drive containing the current build of the game. I found this way of updating suitable as I could control the version of the game that was available to be played ensuring there are no bugs, issues or unfinished things left in the game.

3.4 Choosing technologies

Upon beginning the project, I researched a variety of games to see what engine and language they were created in and found a lot of high-quality virtual reality games that were created using Unity and C#. I also searched for tutorials and guides I could follow. I quickly discovered some small YouTube channels that provided me with some great tutorials on the basics of virtual reality. Channels such as 'Valem'[5] and 'Dave / Game Development'[7].

There were also many guides on sites such as Stack Overflow for bug fixes.

I then researched different ways of developing the virtual reality game. I looked at the oculus virtual reality development kit and the XR Interaction Toolkits and decided to use the latter due to being more well documented and having many guides available.

Chapter 4

Technology Review

There were many technologies used throughout this project.

4.1 XR Interaction Toolkit

The XR Interaction Toolkit package is a high-level, component-based, interaction system. It provides a framework that makes 3D and UI interactions available from Unity input events. The core of this system is a set of base Interactor and Interactable components, and an Interaction Manager that ties these two types of components together. It also contains helper components that you can use to extend functionality for drawing visuals and hooking in your own interaction events.

I imported the XR Interaction Toolkit into Unity using the package manager window. For this project I used version 0.9.4. I used it in order to create a virtual reality camera rig that handles room-scale VR experiences to allow the player to move around freely, for basic canvas UI interaction with XR controllers, Basic object hover, select and grab, and visual feedback to indicate possible and active interactions.

In order to allow the player to interact with game objects, I apply the XR interactable script to that object. This allows the player to grab specified objects, hold them and move them around. I also needed to set custom transforms for each object that can be held so they would be positioned correctly in the player's hand.

There's an abundance of documentation available online through Unity's documents, stack overflow and through YouTube videos. This made it easier to set up and understand and makes virtual reality more accessible to

developers who are interested.

4.2 Oculus Quest 2

The Oculus Quest 2 is a light, compact and comfortable headset and it can be used in a wireless and wired mode. It has many impressive components and specifications such as a powerful Qualcomm Snapdragon XR2 CPU and a 90 hertz refresh rate display. When it is plugged into your computer, it uses your computers components to run and play games and applications. Depending on which version of the Oculus Quest 2 you get, you will either get 64 gigabytes or 256 gigabytes of storage.

This headset is especially impressive due to its low price in comparison to its more expensive competitors. For example, a valve index costs approximately 1100 euros which is a 650 euro price difference between them.

4.3 Unity 3D

I used Unity 3D for my game engine, and this was for many reasons. I have been learning Unity in college with my lecturer, Damien Costello. He taught us the basics of Unity 2D at first with simple movement, player interacting with other game objects through collisions and key presses. He then taught us Unity 3D at the start of my final college year.

Due to being relatively new to game development, Unity is a good option. It is used in many reputable virtual reality games such as Beat Saber. It is also used by a vast amount of indie developers and mobile game developers. It's free of charge, making it even more widely available for students and young people alike. It also has several tutorials on how to make several different types of games and an official website for tutorials too [8]. There are also a vast range of tutorials all over YouTube for many different aspects of games such as game mechanics, game optimizations and many more. Many of these tutorials have helped me with many projects including this one. I have used many different reputable sources for game tutorials such as Valem[5] and Brackeys[9].

I also chose to use Unity 3D as I would like to become a game developer. Many game development companies use Unity for their applications and many companies are seeking Unity game developers.

4.4 Trello Board

During the project I used a trello board to keep track of what I had done and what I needed to do [10]. I created several different boards for different aspects of the game. For example, I had a board for the climbing game and one for the zombie game also. I initially added all my tasks to the to do list. I then made another list for tasks I was in the process of completing and another for completed tasks. Finally, I had a list for tasks that I was having issues with so I could come back to them at a later date.

I found the trello boards useful as it kept me on track throughout the course of the project. It also allowed me to view my progression through the project as more and more tasks were added to the completed list.

4.5 Google Drive

Google Drive is a cloud-based storage solution that allows you to save files online and access them anywhere from any smartphone, tablet, or computer. You can use Google Drive to securely upload files and edit them online. It also makes it easy for others to edit and collaborate on files if you allow them access. Due to the large size of my project, I uploaded the package and build of my game to google drive. This is as it allows for up to 15 gigabytes of free storage space. You can purchase more storage if needed. For example, you can purchase 100 gigabytes of storage for 20 euros per year.

4.6 GitHub

I used GitHub throughout the course of the project as it is an excellent way of keeping track of what work I have done. It can be used to remove unwanted commits, for version control, as a portfolio etc. I had some issues with it at first due to the size of my project. I added a gitignore file to filter out any assets or files that were too large to be uploaded but it was still too large. I tried using LFS or large file size for GitHub which allows up to one gigabyte of storage space for a repository, but it still did not work. I ended up removing several folders form the repository and made them available on my google drive through my package and build files.

4.7 C#

C# is widely used for developing desktop applications, web applications and web services. It is considered a component-oriented programming language. It is the language used by Unity because it is a general purpose language that is so versatile that it can reasonably be used as a scripting language in a game engine I enjoy using C# as it is a simple yet powerful language. C# is an excellent language for beginners as the syntax is fairly similar to Java and there is extensive documentation through forums, official documents and YouTube.

4.8 Visual Studio Code

Visual Studio Code is a streamlined code editor with support for development operations like debugging, task running, and version control. It provides tools a developer needs for a quick code-build-debug cycle and leaves more complex workflows to fuller featured IDEs, such as Visual Studio IDE. I used Visual Studio Code throughout this project as it is my preferred text editor for developing. It is a quick and lightweight editor with extensive add-ons such as compilers, text formatters or even themes.

Chapter 5

System Design

5.1 Virtual Reality Controls

The player is able to freely move about the scene and interact with things due to several scripts.

I got XRRig.cs from the XR Interaction Toolkit. It sets up the VR Rig for the player and allows you to set the camera height for a stationary tracking space. You can set its tracking mode origin which in this case is the floor so it sets the transform to 0,0,0. It also sets the virtual reality camera which is also the players head. The values that are set by it are used in the continuous movement script to control the player further.

ContinuousMovement.cs uses the input values from the input devices or controllers to allow the player to move. It uses the left controllers analog stick for movement and the players head rotation to dictate which direction is forward. I also added a jump method to this script that I am not currently using but in the future I plan to implement it. The SnapTurnProvider script then allows the player to use the right analog script to turn left or right by 30 degrees each turn.

The climber script and the climbInteractable script allow the player to climb when grabbing objects that are using the climbInteractable script. Upon the player grabbing a climb interactable object, the climber script disables the continuous movement script. This is to prevent the player from walking whilst climbing. To allow the player to climb, the controllers velocity is used to detect movement in any direction.

5.2 Enemy Path finding

In Zombie survival, once spawned the zombies will move toward the player's position. This is done using a nav mesh agent component. This component is attached to a mobile character in the game to allow it to navigate through the Scene using the NavMesh specified. I also use it to set a variety of variables such as the enemies height, their turn speed and acceleration. I made sure to set the enemies radius just right as the smaller the radius the closer the enemies can go to walls and objects. If their radius was too large, they would, in some cases not be able to reach the player in specific areas.

Originally the zombies would patrol the area and search for the player but upon changing from the horror themed game, I changed it so that the zombies just go straight to the player, regardless of the players position or whether they're in line of sight or not.

Chapter 6

System Evaluation

6.1 robustness

In terms of robustness, my project is quite good. The player movement, grabbing and camera all function correctly. However there are several bugs in the game that effect game play. Some are merely graphical where as some are game breaking.

The pistol magazines, when loaded into the gun will not stay inside the gun. Upon the player moving the gun or walking around, the mag will sway outside of the gun. The gun still functions as purposed where you can remove and insert other magazines and also fire bullets from the magazines. This issue is a graphical only bug.

A more major issue is the zombies. Every so often, enemies randomly will become invisible will also randomly reappear again. This is due to their mesh glitching below the ground. The zombies however can still be damaged, and they can also still attack the player. This issue makes the zombie level difficult to play.

The player has the ability to walk through walls. This can be done by walking against a wall or barrier and then walking in real life. This issue could be fixed by having the player clamped within a certain X and Z axis grid. This would disallow the player to escape the play area but does not stop them from walking through game objects.

Due to being a solo developer, I had a restricted amount of time to work on every aspect of the game. Many large issues have been fixed through the

course of this project but unfortunately, I had no time to fix the above issues before submission. I believe this is due to my scope being too large for a solo developer. However, I plan to fix these bugs in the future as I wish to publish Monkey Business to a public market place such as Steam.

6.2 performance

When I began the development of my project, the frames per second or FPS was quite low due to several reasons. When the enemies try find the player, I at first used the FindWithTag method. This method searches for a game object that has a specified tag which, in this case was the player. I had this in the update method which runs by default every 0.02 seconds. As a result, the performance was very low. The game was averaging at approximately 40 fps (frames per second). In order to fix this, I only used the FindWithTag method in the start and awake methods that run when the code is initially ran. In these methods I set some variables equal to the found game objects so I could call them instead. I also tried to minimise the amount of times I used FindWithTag as it uses reflection and instead use the GetComponent method

I further improved my performance by setting any game objects that do not move to static, increasing my terrains pixel error count and render distances. However, when I added occlusion culling, my fps decreased drastically. Occlusion culling works by not rendering game objects not in the players view. I am still unsure to why this was happening, but I removed it again from the project. I also tweaked some of the lighting settings, so they are not being continuously rendered. As of the current state, the game runs at approximately 90 frames per second on my computer.

6.3 Limitations

6.3.1 Graphics

I met many limitations throughout the development of my project. Due to my art styles being too realistic there was an extremely large amount of verts or polygons in the scene that caused large frame drops regularly. This made the game run at 45 FPS on average which is far too low for a virtual reality game and can cause motion sickness. I attempted to reduce this by using LOD groups and occlusion culling but it was not enough to fix the issue. As a result, I changed my games art style and theme by using low poly assets

making the game much smoother and easier to run.

After changing the assets, optimizing code and lighting settings, the fps is much more stable now. I get an average of approximately 90 frames per second throughout the game.

6.3.2 Testing Space

When playing virtual reality, the user is recommended to have 2 meters by 1.5 meters of free space for their play area. This is to avoid the user from colliding with walls, tables or any other obstacles. Throughout the development of my game I unfortunately did not have much space to test the game. My play area was approximately 1 meter by 1 meter of space. As a result, I had to be very careful when testing as to not damage any of my hardware or hurt myself. This caused me to take more time when testing the game.

6.3.3 Devices

When developing this game, due to using an oculus quest 2, the game will only function as intended when using oculus hardware. This is as the scripts for the players hands in game are looking for oculus remotes as an input device. Therefore, if the player attempts to play using a HTC vive or a valve index, their hands will not function properly.

6.3.4 Computers

When testing, I used only my computer that has very high specifications due to having high quality, powerful components. I do not have access to any other machine with different components to test the minimum specifications or the recommended specifications.

Chapter 7

Conclusion

To conclude this document, I thoroughly enjoyed creating a virtual reality game using Unity. However, there are still some aspects that I would change if I were to redo the project now.

I faced many challenges throughout this project. Due to being a solo developer, there was a large workload that had to be completed. There were levels to be designed, colour themes to be selected, art styles, game play mechanics to be created and more. I would change this by having another person to work on the project with me in a group. This way we can cover more ground quicker and it also prevents less details to be missed. Another downfall of being a solo developer is time pressure. I didn't have enough time to perfect all aspects of the game such as the enemies becoming invisible and the weapons magazine not staying still inside the gun.

In terms of completing objectives, I believe I successfully completed the majority of them. The game name Monkey Business is a very suitable and appropriate name for the game. I asked friends and family members for their opinion on the name and they deemed it suitable. The game logo is also a very fitting logo as it is simple yet effective. The contrasting colour scheme of black and yellow is colourful and catches people's attention. In terms of the game's art styles, they have changed drastically throughout the development of the game. They began as detailed and realistic models that had dark, grim colours. This was due to the game originally being horror themed. After update 1.3, the realistic aspect was removed due to being too CPU intensive, and was replaced with low poly models, assets and textures. They are stylized and use bright colours to bring a happy, warm atmosphere with them. These assets and models were perfect for the games theme that is family friendly. The distortion effects in the game were ultimately removed.

Although they worked perfectly as intended, they did not suit the new theme of the game. I still however have the effects and scripts to control them in the case that I will have a use for them in the future. The target audience also changed along with the theme. The game was originally a horror game that had a slight gore element to it due to blood effects, grim colours and darkness and was targeted at mature audiences. The theme was soon changed in the 1.3 update and the target audience is for all ages.

The player movement has met its objectives. It is a continuous movement system that allows the player to freely move around using the left analog stick. The player can also rotate left and right by 30 degrees using the right analog stick. The player interactions are smooth as wanted and upon interacting with items the players hand used to interact will disappear and seemingly becomes the item. The Enemy AI works as intended besides the invisibility glitch that I unfortunately didn't have enough time to fix. The players weapon functions as purposed, however the magazine drifts outside of the gun when held. Again, I did not have enough time to fix this, however theoretically by setting the magazine as a child of the gun object when inserted, it would move with the gun.

The Main menu area was created as intended. There are interactable menus, a high score board and an open area for the player to play and practice in. The day and night cycle was completed as intended also. However, its use was reduced to being for aesthetics rather than having a function due to removing the distortion effects.

When I began this project, I was over ambitious and added many different game play mechanics and features. I however managed to complete all of my objectives successfully. However, there are a few glitches as mentioned in the robustness section that I had unfortunately not enough time to complete. This was due to having a scope too large and being a solo developer. I believe that this is a good learning experience for me for the future as I now know what I am capable of doing throughout a certain time period. Therefore, I will be able to make a more suitable scope for myself.

In the near future I will continue to improve my virtual reality game through bug fixes and optimizations initially. I then plan to add more game modes. Currently I have two ideas planned out. I will create a maze game using a recursive backtracker algorithm where the player will have to try find their way to the finish line. They will have a menu on their hand in order to quit the level and give up. I also plan to add some carnival type games such as a

ball toss where the player must get the ball to land in a basket. The score will set through the amount of consecutive balls that are landed inside the basket successfully. Upon missing, the score will restart and either they will have beaten the high score or not. I also hope to add back in a large open world area for the player to roam freely around and potentially implement a driving system into the game so the player can navigate the terrain quicker. Once all bugs are removed from the game and I have added more game modes, I will publish Monkey Business onto a market place such as steam or oculus.

Appendix A

Appendices

A.1 GitHub

Click here for the GitHub Repository! or here's a link
<https://github.com/BurkeBlaine1999/Final-Year-Project>

A.2 Installation

A.2.1 Build File

1. Download the build of the game from my google drive through the link in the projects README or click [here](#).
2. Unzip the downloaded file.
3. Run the 'VR Game.exe' file to start the game!

A.2.2 Unity Package

1. Download the package of the game from my google drive through the link in the projects README or click [here](#).
2. Unzip the downloaded file.
3. Drag and drop the package into an open Unity Editor.
4. Download and install XR Development Toolkit version 0.9.4 from Unity's package manager.

Bibliography

- [1] H. Tankovska, “Forecast augmented (ar) and virtual reality (vr) market size worldwide from 2016 to 2020,” 2020.
- [2] Statista, “Virtual reality (vr) gaming revenue worldwide from 2017 to 2024,” 2021.
- [3] R. F. K. Laura Aymerich-Franch and J. N. Bailenson, “The relationship between virtual self similarity and social anxiety,” 2014.
- [4] B. Vector, “Low poly brick houses.”
- [5] Valem, “Vr game development youtube channel.”
- [6] TurboSquid, “Turbosquid.”
- [7] Dave/GameDevelopment, “Game development youtube channel.”
- [8] Unity, “Unity learn.”
- [9] Brackeys, “Game development youtube channel.”
- [10] Trello, “Trello board.”