



Lighting and Rendering in the Unity Game Engine

FINAL YEAR REPORT

Awais Tasleem SGB849 @00514972 | Final Year Project | 3/7/2023

Abstract

Lighting is a key aspect of any game today and players expect good lighting in these games. The lighting techniques can affect the players' decisions such as in exploring the game world and can emphasize objects such as doors and stairs. Another aspect of today's gaming industry is the amount of protection needed from security threats that affect games.

This dissertation and project show how developers manipulate players' decisions by using certain lighting techniques to dictate the direction the player goes. This was done by making a lighting demo in the Unity game engine and having different scenes to show off the different techniques. One level uses the baking technique which generates a light-map, and it is used like a texture. The other technique is using real time lighting to show how lighting changes with the players' movement and interaction. The lighting demo has been tested by people, and they have stated that the project meets the objectives, and it is a successful lighting demo as it shows how lighting is used in the Unity game engine.

The dissertation also shows what the threats are in the gaming industry and how to protect yourselves against them. This is shown in the Unity game demo with a level that informs the player of the different security issues by using security stories from news outlets and then gives some tips on how to protect against these security threats.

Table of Contents

Abstract.....	1
Table of Figures.....	6
List of Tables.....	8
Chapter 1 - Introduction	9
Motivation	9
Overview of Report.....	9
Chapter 2 - Literature Review.....	10
Introduction.....	10
Lighting	10
Rendering Pipeline in Unity.....	11
Application	11
Geometry.....	11
Rasterization	11
Rendering process.....	11
HDRP vs URP	11
DRM and Security Issues.....	12
Chapter 3 - Methodology	14
What is METHODOLOGY?	14
Agile	14
Scrum.....	14
Programming Tools	14
Unity Game Engine.....	14
C#.....	15
Visual Studio IDE	15
Visual Scripting.....	15
Summary	15
Chapter 4 - Requirements.....	17
Requirements	17
Requirement 1 - Player Movement and Interaction	17
Chaining Moves.....	17
Momentum.....	17
Trajectories and Physics	17
Environmental Integration.....	18

Decision Making	18
Requirement 2 - Lighting Effects and Shadows	18
Requirement 3 – Graphics	18
Requirement 4 – Game/Map Design.....	19
Requirement 5 - Security Features.....	19
Chapter 5 - Design.....	21
Use Case Diagram.....	21
UML Sequence Diagram	22
UML Class Diagram.....	22
Wireframe of UI	23
Summary	28
Chapter 6 - Development and Implementation	29
Sprint 1 – MVP for Demo	29
Starting the Unity Project.....	29
Movement.....	30
Blender.....	31
World Creation	34
Terrain.....	35
Sprint 2 – Final Release	37
Level 1.....	37
Level 2.....	46
Level 3.....	51
GTA 5 Online hackers	51
Call of Duty: Black Ops 3 Remote Access Trojans	51
CD Projekt Red Gaming Source Code Leak	52
Sony PlayStation Live Hack	52
User Interface.....	54
Security Research on IL2CPP	57
Issues During development	58
Technical Issues	58
Personal Issues.....	59
Summary	59
Chapter 7 - Testing and Analysis	60
What is Testing in Game Development	60
Functional Testing.....	60

Usability Testing	60
Compatibility Testing	60
Performance Testing	60
Play Testing	60
Why is testing important?.....	60
Test Plan.....	61
Functional Testing	61
Usability Testing and Play Testing	62
Compatibility Testing and Performance Testing	62
Questionnaire and Results.....	64
Summary	70
Chapter 8 - Critical Evaluation	71
Objectives Met and Deviations	71
Lessons learnt.....	72
Reflections – Deliverables.....	72
Summary	72
Chapter 9 - Conclusion	73
References	74
Appendix a: Project proposal	0
Appendix b: Logbook	4
Appendix c: PlayStation 5 Box	33
Appendix d: Call of Duty: Modern Warfare 2 2022 Battle Pass page:	33
Appendix e: Battlefield 2042 Battle Pass page:	34
Appendix f: Shark Cash Card prices and link to the website - (rockstargames.com)	34
Appendix g: GTA 5 Mod Menu UnKnownCheats - Multiplayer Game Hacking and Cheats - Spectrum FREE GTA V Menu	34
<u>Appendix h: Minecraft Signpost</u>	35
<u>Appendix i: Black Ops: Cold War Dossier</u>	36
Appendix j: Figma Downloads Web Design App for Desktops & Mobile	36
Appendix k: Draw.IO Link diagrams.net.....	36
Appendix l: Jira Support chat 1	37
Appendix m: New AMD Ryzen CPU	38
Appendix n: Construction work pictures	39
Appendix o: Unity Errors in University	40

Table of Figures

Figure 1: Use Case Diagram	21
Figure 2: UML Sequence Diagram.....	22
Figure 3 - UML Class Diagram.....	23
Figure 4 - Overall View in Figma Studio	24
Figure 5 - Main Menu	24
Figure 6 - Play Menu	25
Figure 7 - Free Roam	25
Figure 8 - Demo Mode	26
Figure 9 - Settings Menu.....	26
Figure 10 - Graphics Submenu	27
Figure 11 - Sound Submenu.....	27
Figure 12 – Logs.....	28
Figure 13 - New Unity Project	29
Figure 14 - playerMovement.cs.....	30
Figure 15 - mouseLook.cs.....	31
Figure 16 - Empty Blender Project.....	32
Figure 17 - Low Poly House.....	32
Figure 18 - Low Poly Tree.....	33
Figure 19 - Low Poly Fire Hydrant	33
Figure 20 - Low Poly Traffic Light.....	34
Figure 21 - Gameplay.....	34
Figure 22 - Heightmap	35
Figure 23 - Overview of Terrain	35
Figure 24 - Zoomed in view of Terrain	36
Figure 25 - Vegetation	36
Figure 26 – Corridor.....	37
Figure 27 - Level 1 Movement Camera.....	38
Figure 28 – LockMouse.cs	38
Figure 29 – UnlockMouse.cs	39
Figure 30 - Flashlight corridor	39
Figure 31 - FlashlightOnOff.cs.....	40
Figure 32 - Area Lights and Spotlights	41

Figure 33 - Interior Area Lights	41
Figure 34 - Interior Spotlights.....	42
Figure 35 - Decision 1.....	42
Figure 36 - Decision 2.....	43
Figure 37 - Decision 3.....	43
Figure 38 - Before Baking Orange Light	44
Figure 39 - Before Baking orange and Green	44
Figure 40 - Baking CPU and GPU.....	45
Figure 41 - Lighting Settings.....	45
Figure 42 - After Baking with All lights	45
Figure 43 - Level 2 Map.....	46
Figure 44 – Model Parts.....	46
Figure 45 - Parts in Scene.....	47
Figure 46 - Level 2 Player.....	47
Figure 47 - Temple Lighting	48
Figure 48 – Snakeheads	48
Figure 49 - Door Spotlight.....	49
Figure 50 - Pillar Area Lights.....	49
Figure 51 - Level 2 Lighting and Settings	50
Figure 52 - Before Lighting	50
Figure 53 - After Lighting	51
Figure 54 - Security Room	52
Figure 55 - E Interaction	53
Figure 56 - interact.cs	53
Figure 57 – Dossier.....	54
Figure 58 - Main Menu	54
Figure 59 - Animation Assets.....	55
Figure 60 - Keyframes for Animation	55
Figure 61 – Movement UI	55
Figure 62 - Other Keys UI	56
Figure 63 - LoadScene.cs	56
Figure 64 - Quit Button	57
Figure 65 - Melon Loader	58

Figure 66 - Mono and IL2CPP s.....	58
Figure 67 - Build Tests	61
Figure 68 - AMD Laptop Error	63
Figure 69 - Game running on University Computers.....	64
Figure 70 - Questionnaire Part 1.....	65
Figure 71 - Questionnaire Part 2	66
Figure 72 - Questionnaire Part 3.....	67
Figure 73 - Results part 1	68
Figure 74 - Results part 2	69
Figure 75 - Results part 3.....	70

List of Tables

Table 1 - Objectives.....	71
---------------------------	----

Chapter 1 - Introduction

The main aim of this project is to understand how lighting and rendering works in games and how it can help the gameplay and atmosphere for the player by making a lighting demo project. I will also try to learn how the new Unity Render Pipeline works for lighting works, and how you can optimize games performance so that my game can run on lots of older and slower hardware such as older graphics cards.

Lighting will be used to enhance the gameplay visually by using it to direct the player to a certain place, or to emphasize an object such as a door or hallway.

Another part of the project show the security aspect of the gaming industry as it is becoming a bigger issue as more games are getting hacked and gamers details are being leaked such as the Sony PlayStation online hack that happened in 2011 made Sony lose \$171 million (Mystic, 2019).

The approach for this project will be using agile methodology to develop the game demo quickly and efficiently. It will be broken into 2 main parts, the MVP, and the final release. The final release will have 3 parts which are the 3 levels.

MOTIVATION

I wanted to make a game with good graphics for a long time as I have made games in the past that are on the google play store and on my website [Awais Tasleem's Portfolio \(awaist7860.github.io\)](https://awaist7860.github.io). The main issue that I had when developing my games was the time needed to make them as I was in high school and college studying for my GCSE's and A-Levels, I did not have enough free time to make the games. This project will allow me to spend more time on learning more about game development and the different practices that are used in the industry and to improve the skills that I have already.

OVERVIEW OF REPORT

The report starts with the literature review which talks about how lighting is used in Unity and the importance of lighting in games. The next part will talk about the methodology chosen, which is the agile methodology. The next part will talk about the specifications of the project and what designs and design tools were used. The next part will show the development and implementation of the project from start to finish. The next part will talk about the testing and what testing methods were used. The next part will critically evaluate the project and the first 2 deliverables. The final part will conclude the work and give suggestions for future work.

Chapter 2 - Literature Review

INTRODUCTION

This literature review discusses how lighting is used in the 3d game engine, and how important lighting is in any game and what lighting can show the gamer either physically in the game world, or emotionally in the gamer's mind. It will also discuss how lighting in game development has changed and improved over the years and how it has changed the industry, for example ray tracing has made a huge impact on games in the past few years where even the new generation of consoles (PlayStation 5 and Xbox series consoles X | S) and phones with the snapdragon 8 gen 2 SOC (System on a chip) now focuses on ray tracing and lighting after Nvidia released the RTX series of graphics cards back in 2018. I will also talk about the Unity game engine and what role this engine has played in the game development industry.

LIGHTING

In Unity, Lighting is approximated to how light works in real life. Unity allows the use of models that have high detail to get a more realistic look, or a more simplified model to get a more stylized look. (Unity Technologies, 2022)

Unity uses several different methods for lighting and one of these methods is Direct lighting. Direct lighting is when the light has bounced off a surface once and is reflected to a sensor such as the game camera. Indirect light is all the other types of light that is reflected to the camera such as skylights, multiple bounce lights. Most realistic games use a combination of them both to make a better-looking game.

Another lighting method that Unity uses is baked lighting. Baked lighting is technique of performing lighting calculation before runtime, and then the results are stored as lighting data, then when the game is run, the engine will get the lighting data and apply it at runtime. This saves in lots of performance as the engine only needs to do a lookup operation instead of a full calculation.

Another lighting method that Unity uses is real-time lighting. This is the opposite of baked lighting where in baked lighting the lighting data is calculated beforehand, but for real-time lighting, the data is calculated at runtime, this could be every frame in the game by using the update method in Unity. This way of doing lighting is very taxing on the graphics card as the calculation needs to be run on every frame. Nvidia has a solution to do this by using RT cores in their RTX line of graphics card, where all the lighting calculations are done on those cores, which increases performance.

Another method that Unity uses for lighting is called Global Illumination. Global illumination is a group of techniques from the direct and indirect techniques to make a more realistic looking game. Unity has 2 built-in global illumination systems, which combine direct and indirect lighting. (Unity Technologies, 2022)

One of the global illumination systems is called the baked global illumination system and it uses light mappers which calculate the brightness of the surface before runtime, light probes which provide a way to see data of light moving through an empty space

in the game scene and reflection probes which 'allow the visual environment to be sampled at strategic points in the scene'. (Unity Technologies, 2022)

Unity has a build in project tutorial that shows how lighting works in a HDRP project, and I have followed this tutorial and worked on a sample scene.

RENDERING PIPELINE IN UNITY

Rendering is the process of drawing a game scene which could be full of models and lighting on a computer screen. It involves a combination of many different methods, such as geometry calculations, textures, and lighting. (Schrute, 2019) The rendering pipeline transforms all the game data such as models, lighting, and textures into a virtual environment on the player's screen to show the player the game world. A rendering pipeline consists of 3 layers, Application layer, Geometry Layer and Rasterization layer.

APPLICATION

The application layer is where the developers command the calculations for the geometry and the rasterization. This is the top of the stack. The application usually talks to a graphics API such as DirectX 12 which then talks to the physical hardware on the device.

GEOMETRY

This layer consists of all the shapes and vertices that make the models in a game. This layer also calculates the position of the camera in the game world. The engine then calculates all the data and then passes them onto the rasterization process.

RASTERIZATION

This is the process of converting 3d modeled objects and the environment data such as lighting and textures to a 2d plane. This is since our device's screens are flat, and the data needs to be converted to fit this view or the game would not look correct to the player. The conversion sends the 3d world data through filters that then in turn show us a rasterized 3d image on a 2d screen.

RENDERING PROCESS

The rendering processes consists of Geometry, Illumination, View Perspective, Clipping, Screen Space Projection, Adding Post Processing and Display.

HDRP VS URP

Before starting up a new unity project I had to decide which render pipeline I had to use. A render pipeline performs a set of operations that take the contents of a scene and display them on the screen. (Unity Technologies, 2022) Unity has allowed developers to create their own render pipeline, but they have also provided 2 render pipelines for us to use. One is the universal render pipeline (URP) and the other is the high-definition pipeline (HDRP). A HDRP project is not compatible with a URP project. (Unity Technologies, 2022)

URP is a prebuilt render pipeline made by unity to allow developers an easier and user-friendly way to build their games that will work on a variety of devices such as mobile phones to high end desktop PC's. (Unity Technologies, 2022)

HDRP is a high-fidelity scriptable render pipeline that was made by Unity to target higher end hardware such the new Nvidia RTX 4090 graphics cards. (Unity Technologies, 2022) HDRP utilizes physically based lighting techniques, linear lighting, HDR lighting and a configurable hybrid tile/cluster deferred/forward lighting architecture. (Unity Technologies, 2022)

For this project I have chosen to use Unity's high definition render pipeline, as it allows me to have more realistic lighting, it also allows for real time global illumination which is a group of techniques that model both direct and indirect lighting to provide a realistic lighting effect on the world. (Unity Technologies, 2022)

Another feature of this pipeline that I may use as it is one of my optional objectives, is that it allows me to use raytracing in the game to give a more real world look and feel for the lighting. Raytracing is the process of generating an image by tracing out rays from the camera through each pixel and recording the color contribution at the hit point. (Unity Technologies, 2022). The main issue with ray tracing is that it requires a lot of graphical horsepower to run, and my current system may not handle it well, but I may be upgrading in the near future, and this is something I may be able to focus on which is the reason why it is an optional objective.

DRM AND SECURITY ISSUES

DRM stands for digital rights management, and it is used in game development to reduce piracy of games. If you are buying a game on a digital platform like Steam on PC or PlayStation store on PS5/PS4, you aren't buying the game, you are buying a license to run the game on the specific hardware that the license allows. If you buy a game on the PS5, you won't be able to play that game on an Xbox Series X console.

The DRM software first checks for a license file when the game first launches. If the DRM software finds a match, then the game will run normally, and the DRM software will stop running. If the DRM software doesn't detect the correct license file, then the game will not run, and an error message will be shown to the player.

Some new DRM software requires a constant internet connection all the time and the DRM software runs all the time when the game is launched. This was used on many games such as sim city 2013 and Diablo 3. (Roach, 2020) This can cause performance issues when a game is run, but developers are reducing the use of this method.

Another security measure that games use called the Anti Cheat system. Cheating has become a big threat to digital online games as a cheat could render a game unplayable by players and then the game is abandoned, which would make the development company lose lots of game revenue.

One way a cheater would exploit a game to get a competitive advantage was to modify game drivers to allow them to see through walls in game or have an aim bot that would

give the player perfect aim. These would make other players angry, and they would rage to quit the game. Some of these wouldn't return to the game or they would only play the game with their close friends. Some would completely leave the game and never join again; this would make the games development companies lose lots of revenue. (Ellis, 2020)

A controversial way that companies are trying to stop cheaters in games is releasing anti cheats that access the kernel at ring 0 level. One game that does this is Valorant that uses ring level 0 for its cheating software. This is a huge on pc games as this level is the most central part of the OS, if anything fails here, it will probably cause an entire system shutdown. (Raspberry Pi, n.d.)

Chapter 3 - Methodology

WHAT IS METHODOLOGY?

A methodology is a structured process that is used when working on a project. The main reason for using a methodology is to provide a systematic approach to the software development process.

A definition of a software methodology is 'a framework that is used to structure, plan, and control the process of developing an information system.' (Saylor.org, 2021)

Agile

I have chosen to use the agile methodology during the development of this project. The reason I have chosen the agile methodology is that it allows me to create a feature and then quickly respond to the change. In game development, requirements are always changing, and deadlines are also changing, so that is why agile is used a lot in the game development industry. Agile is for quick iterative software changes on shorter sprints.

All my objectives that are stated in my project proposal are compatible with the agile framework, as each objective can be broken down into smaller parts and then all the smaller parts can be joined together to make the entire project complete. One example of this is that objective 6 is where I need to complete a minimal viable product, and objectives 7 and 8 are improving on objective 6. This shows that adding small quick features makes it easier to work on developing the game instead of working on the project as a whole and then testing at the end.

One major advantage of Agile development is that it allows more collaboration between a team and allows them to work faster together. It allows the teams to work on smaller projects instead of working on the entire project at once.

Scrum

Scrum is an agile methodology, and it is designed for quick short sprints, usually 1 week, where you work on a feature and evaluate it then get feedback on it within a week. This is one of the methodologies that I used when I was on placement last year at Knorr-Bremse.

Scrum allows the developer to quickly add features, and then also test and fix errors quickly. In game development, this is very useful as one feature could break many other features, and quick testing and fixing allows the game to be more smoothly developed and reduced the number of bugs that are in the game.

PROGRAMMING TOOLS

Unity Game Engine

I will be using the Unity game engine to develop my game. Unity is a professional game engine that has been used to make many AAA quality games such as Fall Guys: Ultimate knockout, Among Us, Cuphead and Pokémon Go. Unity is also used to make

many mobile games. About 50% of all mobile games are made using Unity. (Unity Technologies, 2022)

Unity Game Engine has been around since June 2005, and it was a paid software. Then in 2009, the Unity game engine became free to use by anyone, this was announced at the Unite 2009 Conference. (Helgason, 2009)

C#

Unity game engine allows scripting in the C# language which was developed by a Microsoft employee called Anders Hejlsberg. (Pluralsight, 2022) C# is an object-oriented language and makes developing games easier as you can create C# objects that are linked to game objects. For example, you could have a player model and the class linked to it would be a player class, and the class would have sub classes such as player health, player armor and player inventory. These classes would be linked to the actual 3d model in the game.

Unity also allows other .NET languages if they can compile a compatible DLL file, this means C++ could be used for faster and efficient running code in the game as C++ is a lower level, more down to metal language and can run code much faster than C#. (Unity Technologies, 2021)

Visual Studio IDE

I will be using Microsoft IDE called Visual Studio 2022 to code in C#, and these scripts automatically get added to Unity as Unity has built-in support for opening external scripts.

Visual Studio was released in 1997 by Microsoft, and it is an IDE used industry wide by many professionals and companies. When I was on placement last year, I used Visual Studio to do my work. Visual studio is a multi-language IDE where it has support for more than one programming language. You can code in C#, C++/C, JavaScript, python, F# and visual basic.

Visual Scripting

Unity game engine also allows visual scripting like the game engine called Scratch which allows the user to drag and drop blocks of instructions to code the game object. Visual scripting allows easier collaboration between programmers, artists and designers and it allows for faster prototyping and iteration. (Unity Technologies, 2021)

Unity's visual scripting contains logic-nodes attached to each other to do an action, this could be a walking animation where a key press makes the player go forward and an animation if this is shown.

SUMMARY

This project will be developed using the Unity Game Engine. I will be learning to code using Unity's documentation and online tutorials. I will also be using the Visual Studio IDE to code in. The 3D models will be made using the Blender modeling software, and

the textures will be created using photoshop. The Project will be managed by using Trello. I will be targeting my Unity builds to be run on windows.

The main skill that I need to focus on is my time management skill as this project is worth 40 credits and I need to spend an appropriate amount of time on it. It will also allow me to stick to my methodology.

Chapter 4 - Requirements

REQUIREMENTS

This part of the report will talk about what requirements are required for the project and it will also give a reason as to why they are important, it will also give a method to complete the requirement.

Requirement 1 - Player Movement and Interaction

Character movement and interaction in games is a common thing and is usually overlooked by games, but designing a great character movement mechanic can make the gameplay more fun and enjoyable for the user (Run, 2019). If the character movements mechanics are bad, this will annoy players and stop them from having fun in the game.

Movement and interaction mechanics are the main way that allows a player to move around and get immersed in the game world (1-up, 2021). This is very important for any game from the first Mario World game to today's huge 3D open world games such as Horizon: Forbidden West.

Character movement will be an essential part in showing the lighting and Rendering of the Unity game Engine as the player will need to be able to move around the game world so that they can see the way the light is rendered in the game and how the player movement effects the lighting. To achieve this goal, the project will be constantly tested with other people to see if the movement feels good.

Below are 5 factors that are very important to the core of most games currently on the market.

Chaining Moves

Chaining moves together increases the depth of simple key presses (1-up, 2021). The simple action of chaining moves together gives the player more option of exploring the game world and interacting with the game world, and it also rewards more skilled players (1-up, 2021).

Momentum

Momentum is the feeling of speed and non-stop movement for the player. This allows the movement of the game to feel fluid and smooth to the player (1-up, 2021).

Trajectories and Physics

Another important part of having a well thought out movement system is having a realistic or believable physics system. This part allows developers to make the movement system easier for the gamer to understand as it feels like real life to them (1-up, 2021).

One example where the physics of the real world are used to affect the movement system in a game is in the Spider Man game on PlayStation. The player moves around the city of New York as spider by swinging from one building to another (1-up, 2021).

To achieve this goal, the project will integrate gravity and jumping into the demo so that the player can easily move around the world.

Environmental Integration

Using the game world to improve the movement of the game is another excellent way to make the movement mechanics feel solid and enjoyable and make the game more immersive to the player. The world becomes an obstacle and a tool for the player (1-up, 2021).

Decision Making

Decision making involves creating situations where the player is encouraged to think critically about the options that are available to them using good level design and movement options (1-up, 2021).

Requirement 2 - Lighting Effects and Shadows

This requirement is the main aspect of the entire project. The lighting in a game is one of the most important parts of any game as it can affect the look and feel of the game, and it can affect the gameplay of the game as well (Pluralsight, 2014).

Lighting is used to guide the player throughout the game world. A dimly lit room suggests that the room is full of danger to the player, and a well-lit room shows that the room is safe. This also allows the developer to set the tone of the and mood of the game, by making the lighting bright, it implies peace, cheerfulness, and safety. Whilst making the lighting dim, makes the mood to be dark, bleak, and gritty. Lighting can also affect the gameplay by making it easier or harder to see objects.

One reason why lighting is heavily focused on by developers, is because it adds a sense of realism to the game (Alexander, 2018). Good realistic lighting can make objects look more realistic and three-dimensional.

This will be achieved by showing different scenes with different lighting effects to show the user what the Unity game engine can do and how lighting can be manipulated to change the experience of the game.

Requirement 3 – Graphics

This requirement is similar to the lighting requirement above because graphics is what the player sees when playing the game, this includes everything from the environment and the character and it also includes the lighting. (Pluralsight, 2014). The better the graphics of a game look usually makes the game be successful because it attracts players to the game. The games console companies that make the consoles that these games run on do the same thing with Sony's PlayStation 5 boasting that it can play games at 8k resolution or play games at 4k 120fps (Appendix c):. To achieve this

requirement, the models in the games will be downloaded from online asset stores as other people have made higher quality assets and textures.

Good graphics aren't the golden bullet that will increase the sales of a game, for example the game Ryse: Son of Rome, had very good realistic graphics and lighting for the time but didn't receive much positive feedback because the gameplay wasn't up to par with the graphics as the gameplay was boring and repetitive (Pluralsight, 2014).

Requirement 4 – Game/Map Design

Good map design can greatly enhance the gaming experience and make the game feel more fun and enjoyable for the player (TRAINOR-FOGLEMAN, 2021). A well-designed game will have a huge impact on the gameplay, this will affect the pacing of the game, the difficulty of the game and what choices the player can make. A well-designed map can make memorable experiences for the player. Also, a well-designed map will allow the player to easily traverse the game world.

Another way that good map design has a positive impact on the player is it can make the player keep on coming back using a battle pass which is a business model used in many free to play games where the developer gives players rewards for completing in-game milestones. When a player levels up or earns and unlock token, they can choose to unlock the reward (Computer Hope, 2021). Call of Duty: Modern Warfare 2 2022 rewards the player with an unlock token. The player can then decide which reward they want to unlock (Appendix d:). Battlefield 2042 doesn't allow the player to pick the reward, the game will automatically give the reward to the player (Appendix e:).

The project will achieve this requirement by having in game lighting direct the player to the right way in the demo, by having light area and dark areas. The dark areas will suggest to the player that they shouldn't go there, and the light areas will suggest that the player can go in that direction.

Requirement 5 - Security Features

Game companies require players to create accounts such as EA access or PlayStation account to play games online or to even purchase games digitally. The handling of personal data for the game companies is very important so that they don't break GDPR rules and harm the reputation of their brand and get a huge fine. Hackers on the other hand want to get access to this data so that they can either use it themselves to commit identity theft, or they will sell the user data on the dark web.

Hackers will also want to steal game assets such as source code so that they can sell the data to game crackers or game pirates so that they can easily bypass DRM and copyright protection on the games. This has recently happened to a development company called 'CD Projekt RED'. The company was hacked and the game files and source code for the game Cyberpunk 2077 were stolen and later sold on the deep web (Criddle, 2021).

Another game that is full of hackers and cheaters is Grand Theft Auto 5 Online. There is an exploit in the game that allows a hacker to see the IP address of anyone that is connected to that session of the game (BARI, 2023). By using this exploit the cheater can give themselves unlimited in game currency. This affects Rockstar Games revenue because they sell SHARK CASH CARDS that give the player in game currency for real world money. The range is between £3 and £60 (Appendix f:). The cheaters also have the power to reset players' stats which would make them lose all the progress they had in the game. Also, they can give themselves max rank in the game which will make the game easier for them, but it will give the other players an unfair disadvantage. All of this can be done from the Mod menu that the hackers and cheaters have created (Appendix g:).

The exploit also allows the hackers to install Remote Access Trojan on to anyone's PC connected to their game sessions. They could also run ransomware on the victim's machine and lockup and encrypt all their files until the victim pays a ransom (SomeOrdinaryGamers, 2023).

This project will aim to teach the player about the security threats that are in the game development industry by giving the player 4 security stories. This project will also give the player a mission to improve their own security.

Chapter 5 - Design

This section is going to go over the details of the designs used during the development of the project. The diagrams were created using a web application called draw.io (Appendix k:) This section will also show a wireframe look of the UI, and this is made using a software tool called Figma (Appendix j:) which is a tool that is used to create UIs for websites, apps, and other digital products like the UI of a game.

Use Case Diagram

A use case diagram shows the interaction between the user (Gamer) which is called an actor and the system (Lighting Game) which contains use cases. The purpose of the diagram is to show a high-level view of the system functionality so that it is easier for the developers, designers, and stakeholders to see what the project is going to look like and make the requirements of the system more clear as well (Lucid Software, 2018).

Below is Figure 1: Use Case Diagram, which is a use case diagram for this project, and it has identified that there is only one actor which is the gamer. The system has many use cases and many of the use cases have other use cases that rely on the main one. There are 5 main use cases and 3 of the use cases have extended or included use cases added to them.

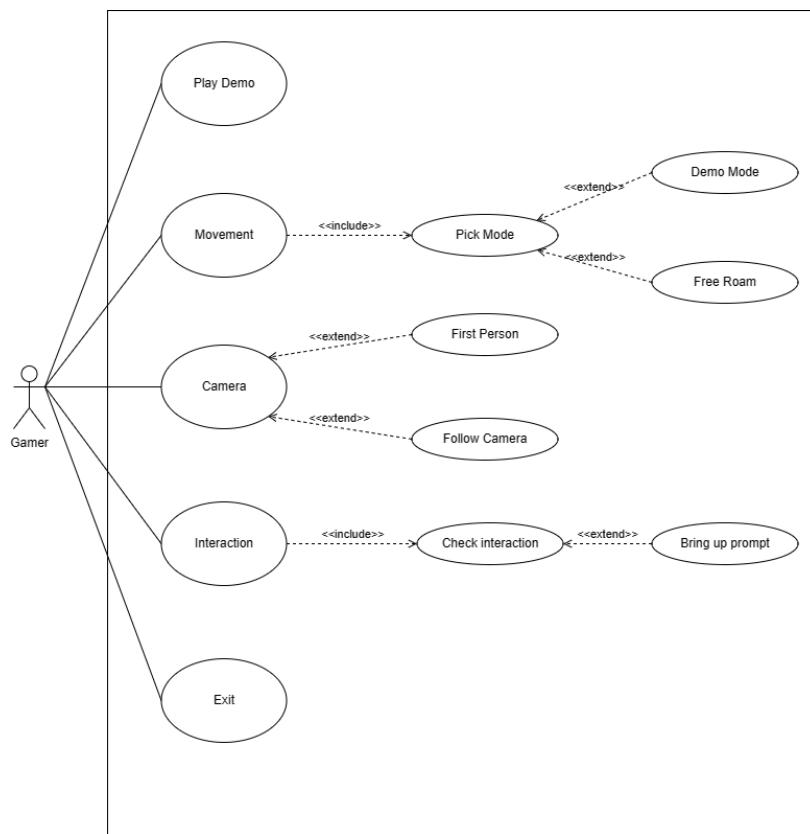


Figure 1: Use Case Diagram

UML Sequence Diagram

A UML sequence diagram shows the sequence of messages between objects in a system. The main purpose of a UML sequence diagram is to show a visual representation of the dynamic behavior of a system (Lucid Software, 2018).

Below is a Figure 2: UML Sequence Diagram which is a UML sequence diagram for the project. It shows the interaction between a game object within the game world. When the player gets near an object that is interactable, a sequence of events will see if the object that the player is near is interactable or not. If it is interactable, then the player should get a prompt of some kind. If the object is not interactable, then nothing should happen, and the game should allow the player to move around the world with no prompt showing up.

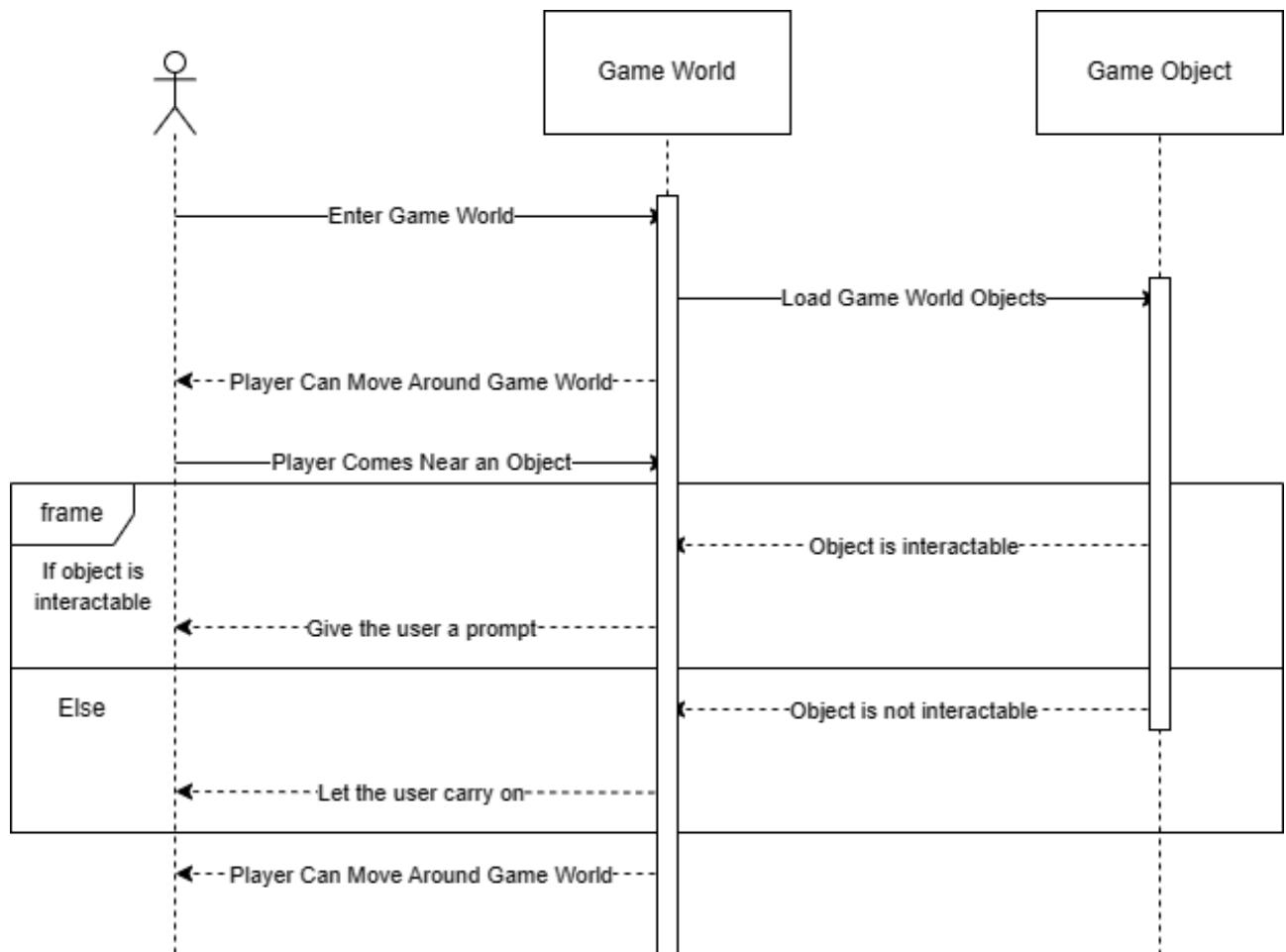


Figure 2: UML Sequence Diagram

UML Class Diagram

A UML class diagram gives the user a visual representation of the classes in the system. It shows the attributes and methods of the classes and relationships between classes. UML class diagrams are a very important tool for designing and documenting a system by helping developers visualize and communicate the structure and behavior of the system to the stake holders and team members (Lucid Software, 2017).

Below is Figure 3 - UML Class Diagram which is a basic UML Class diagram for the project. This most likely will change as the development continues. There are 3 main classes Game, Player, and World. The game class manages the other 2 classes Player and World. The player class is the main character that is controllable by the player. World class is a game world which contains objects. Objects can be interacted with if they are interactable. The camera class is the view in the game and the player sees the game world through the camera. The camera can be controlled to move around and show the game world.

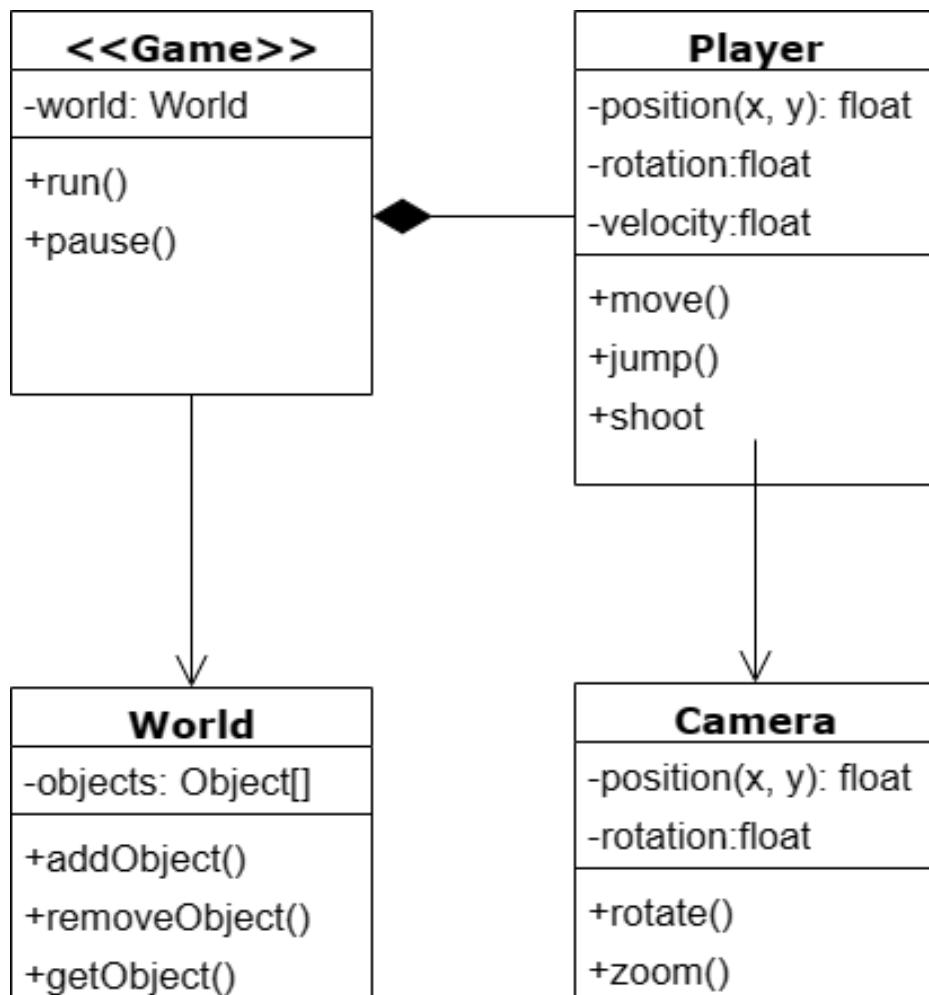


Figure 3 - UML Class Diagram

Wireframe of UI

A wireframe is a visual representation of the layout and structure of an application such as a website, application, or a game. Wireframes are typically created in the early stages of the development process, and they usually change over time as it is just a mockup of the final design. The main purpose of a wireframe is to show the basic outline of a product and to be as simple so that everyone can understand it (Gemayel, 2019).

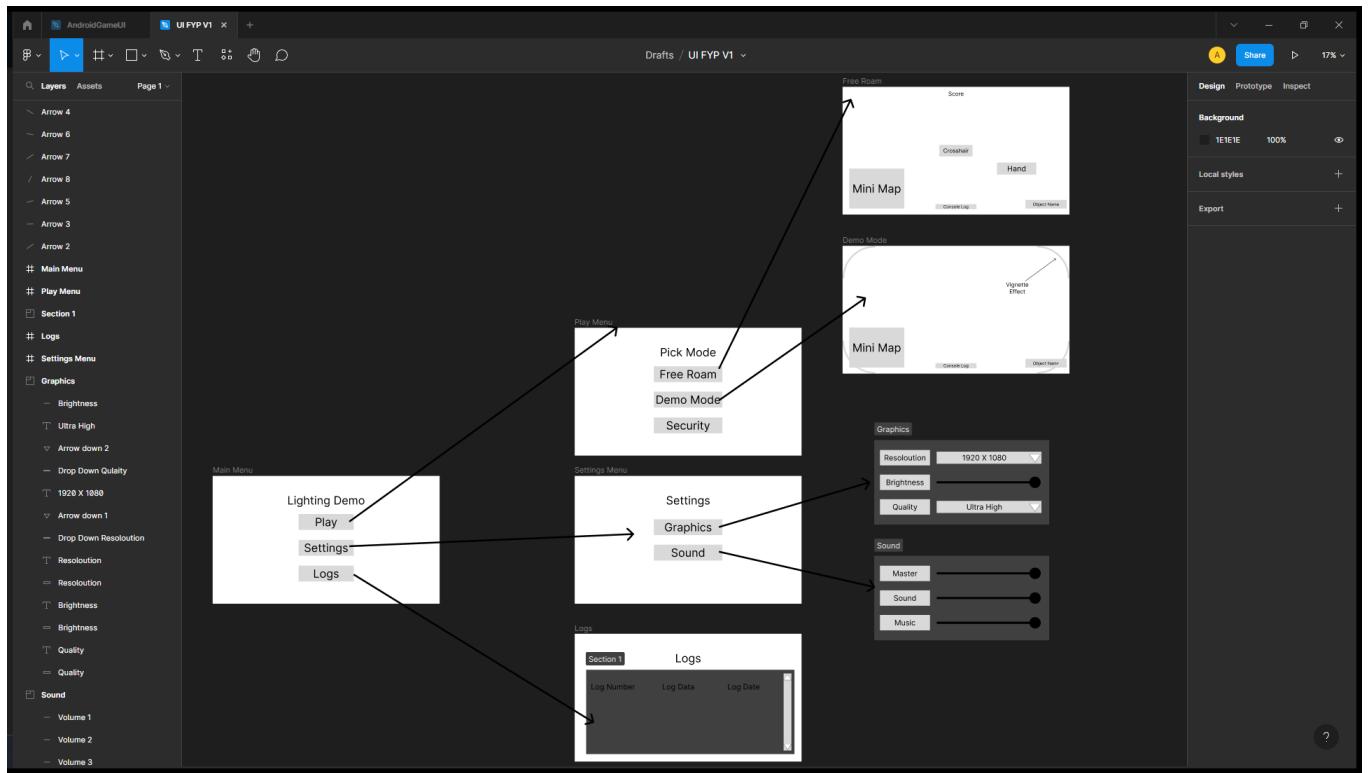


Figure 4 - Overall View in Figma Studio

This is the overall view of the UI. This shows the flow of the UI.

Lighting Demo

Play

Settings

Logs

Figure 5 - Main Menu

This is the main menu of the game, it gives three options, Play, Settings and Logs. The play button will go to the play Menu where the player can pick the mode that they want to go into. The settings button will take the user to the settings menu. The logs button will send the user to the log menu to see any issues that may come up.

Pick Mode

Free Roam

Demo Mode

Security

Figure 6 - Play Menu

This is the play menu. The user has 3 modes to pick from. The Free roam mode will allow the user to explore the entire game world. The demo mode will be a curated mode where the camera will move around the world on its own and shows the lighting effects in the game demo. The security button will show the user a small security demo.

Score

Crosshair

Hand

Mini Map

Console Log

Object Name

Figure 7 - Free Roam

This is the free roam wireframe.

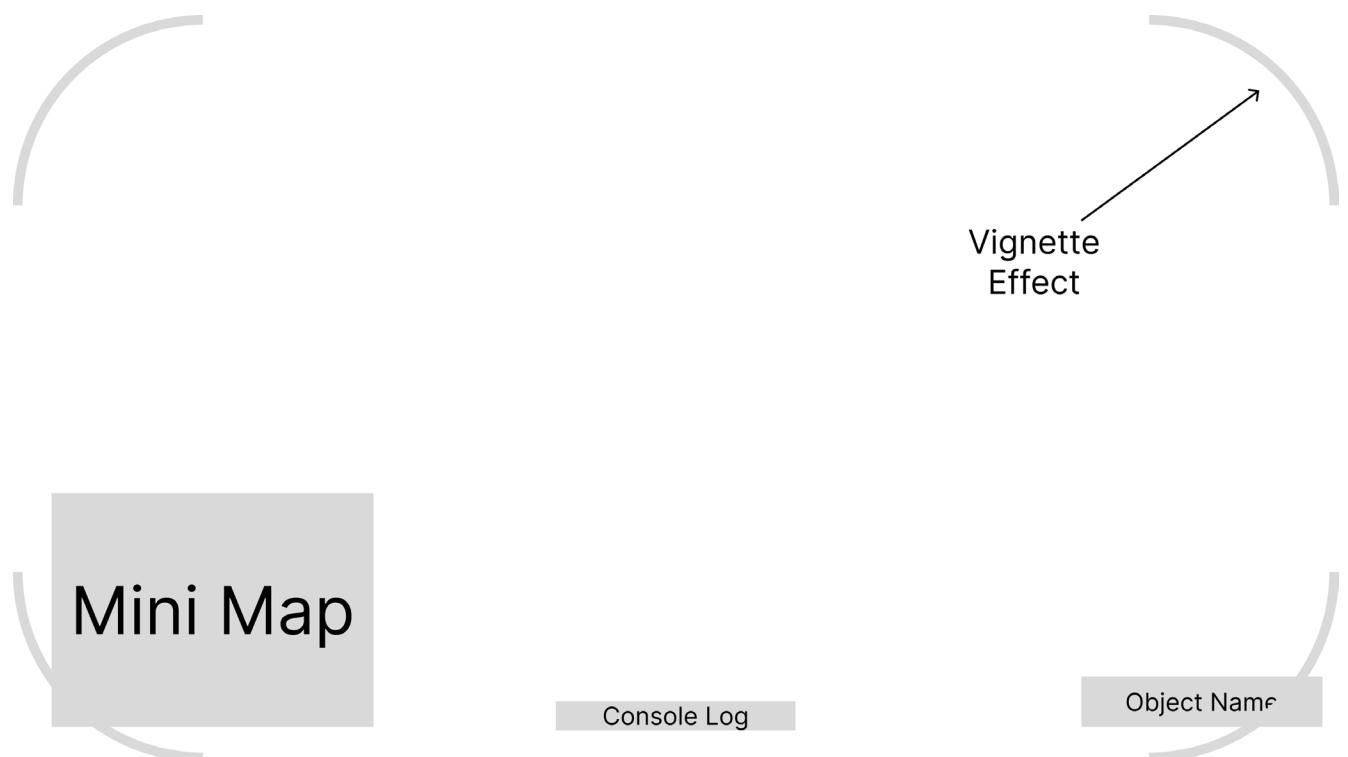


Figure 8 - Demo Mode

This is the demo mode wireframe.

Settings

Graphics

Sound

Figure 9 - Settings Menu

This is the settings menu. When the player clicks on graphics button or the sound button to adjust them.



Figure 10 - Graphics Submenu



Figure 11 - Sound Submenu

Logs		
Log Number	Log Data	Log Date
Section 1		

Figure 12 – Logs

This will be a debugging area so that it will be easier to find bugs and errors that happen in the game.

SUMMARY

The design of the game demo has been simplified to mainly focus on the lighting effects, which is the main purpose of the project. Also, the UI is simple and easy to follow so that the user can quickly get in and out of the different parts of the demo so they can quickly see the entire demo.

Chapter 6 - Development and Implementation

This section is going to show the process of developing the lighting demo project fully from start to end. It will show how the design of the project becomes a fully finished lighting demo project. It will also discuss how the requirements in the previous section are implemented into the project.

SPRINT 1 – MVP FOR DEMO

Sprint 1 is dedicated to showing the MVP for the lightning demo. The MVP will be focusing on getting a build of the project working. The MVP consists of 5 key parts, these are, running unity, developing the player movement, creating a basic game world, creating assets in blender, and creating a terrain.

Running the unity game engine to make sure everything is running. Developing the player's movement so the player can look around the world and move around the world. Creating a basic world so that the player can stand and see the world. Create 3d models in blender so that there are some real assets in the demo. Make terrain so the player can walk on it instead of just having a basic floor.

Starting the Unity Project

The implementation of the MVP was to have a stable unity project started up and running. The first thing that was done was a floor was added to the main scene. After this was done, the game was then tested by building it to a windows platform target.

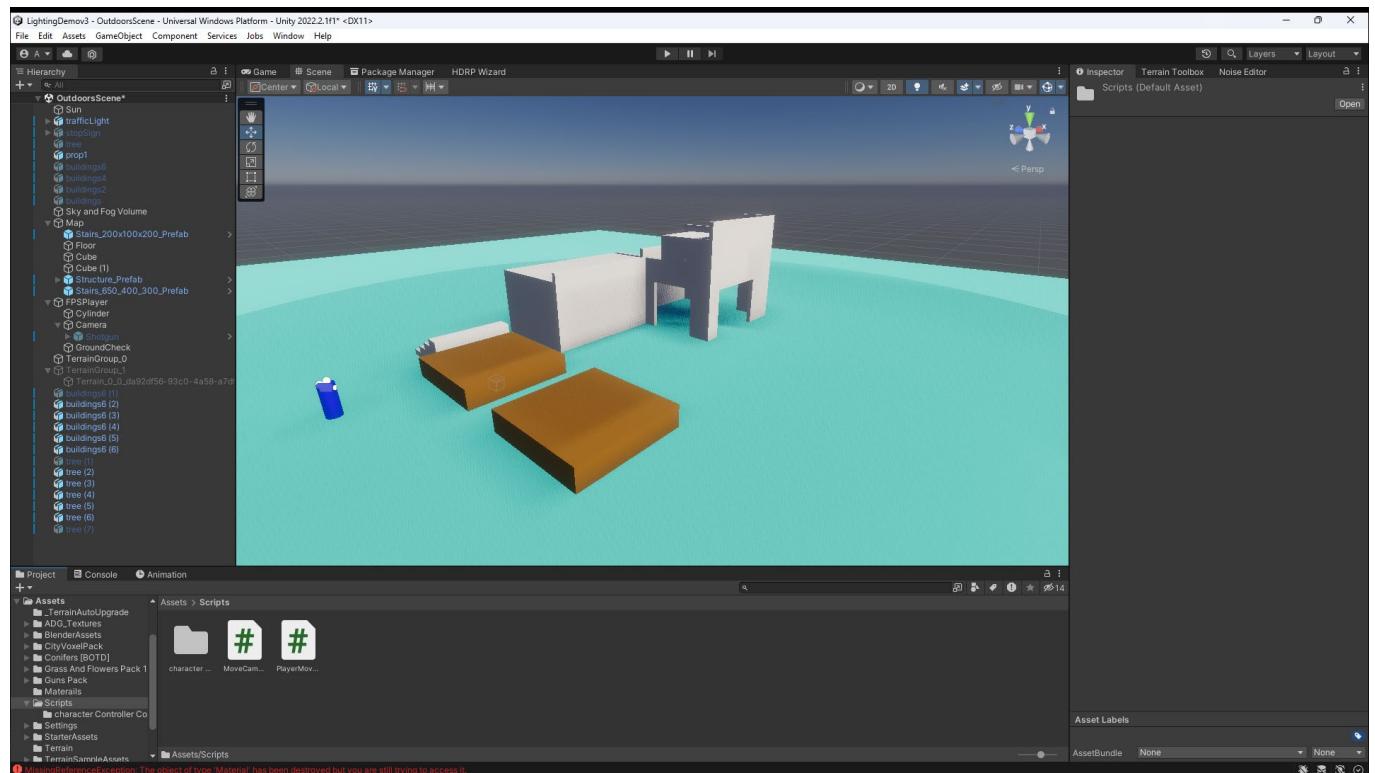
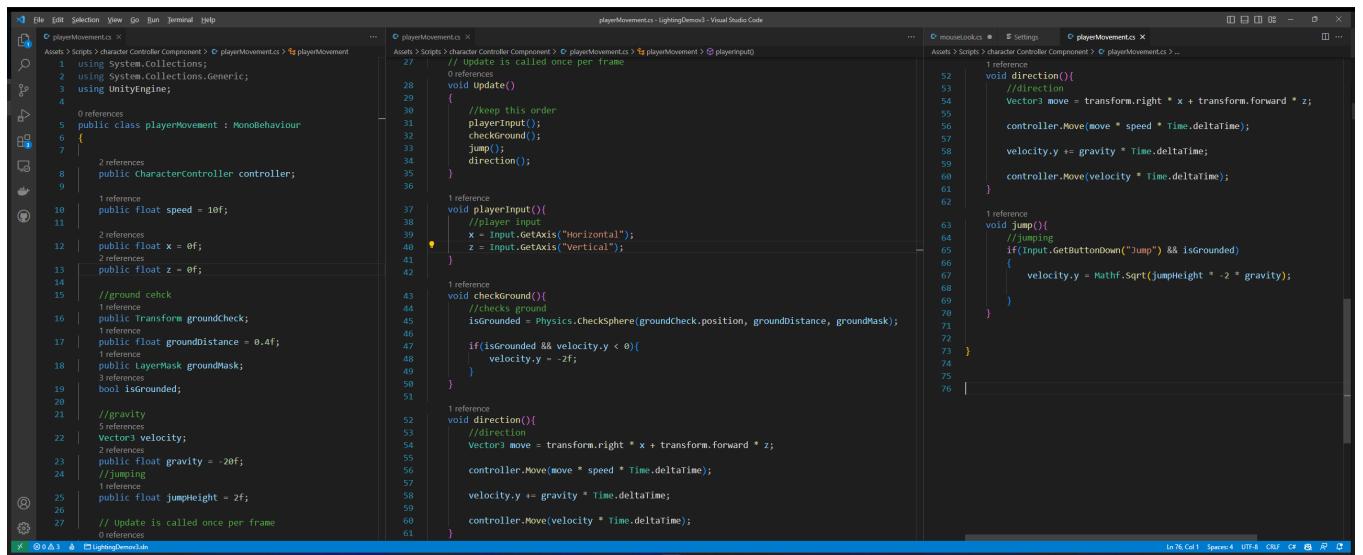


Figure 13 - New Unity Project

Movement

The next step in the development process was to implement the player movement mechanics into the project. This had to be coded in C#. You can also use Unity's new visual scripting tool, but this was not necessary for the MVP. The movement mechanics has a few components, these are movement, gravity, ground checking, jumping, in-air movement, slopes, steps and doesn't get stuck on walls (Brackeys, FIRST PERSON MOVEMENT in Unity - FPS Controller, 2019). Figure 14 shows the movement 'playerMovement' script that allows the player to move left right, forwards, backwards and jump. Figure 15 shows the 'mouseLook' script. The main purpose for this script is to allow the player to look around the world.

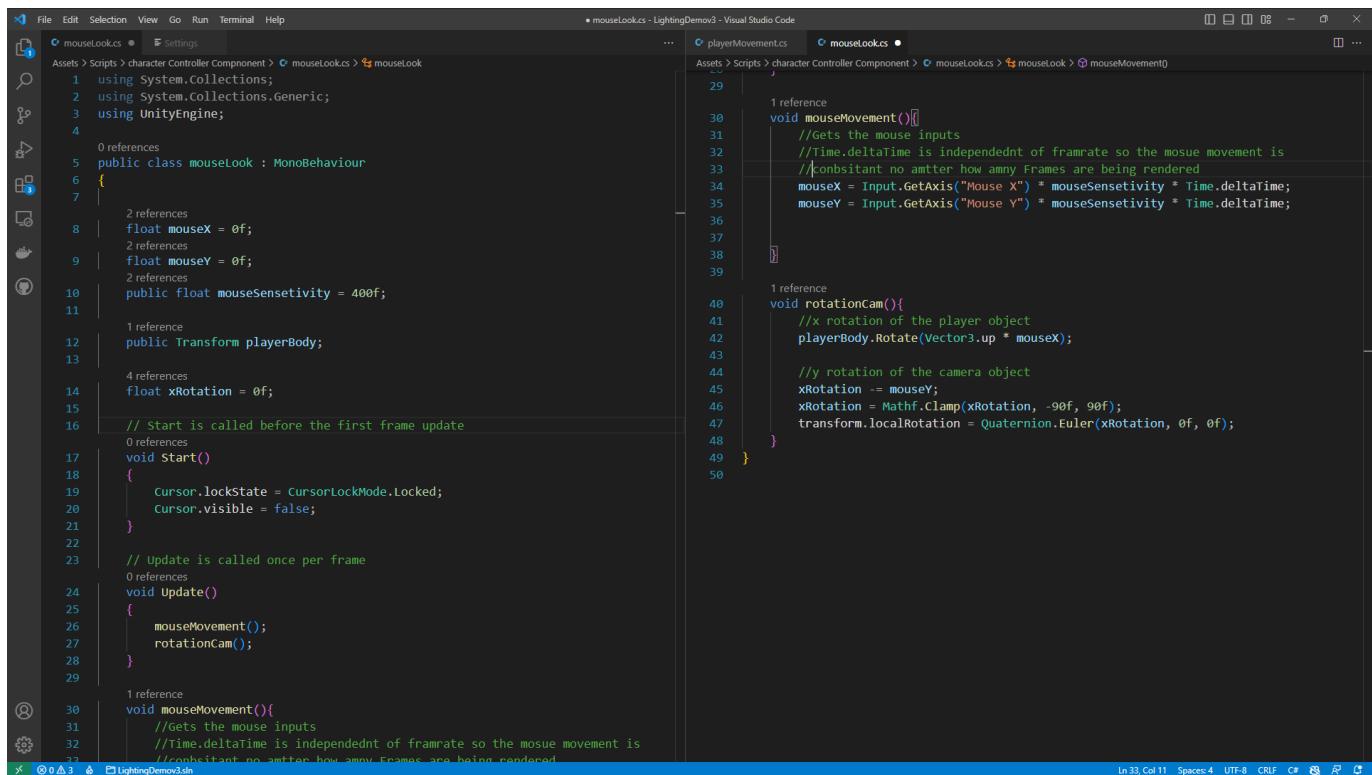


```

playerMovement.cs
Assets > Scripts > characterControllerComponent > playerMovement.cs
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class playerMovement : MonoBehaviour
6 {
7     public CharacterController controller;
8     public float speed = 10f;
9     public float x = 0f;
10    public float z = 0f;
11
12    //ground check
13    public Transform groundCheck;
14    public float groundDistance = 0.4f;
15    public LayerMask groundMask;
16    bool isGrounded;
17
18    //gravity
19    Vector3 velocity;
20    public float gravity = -20f;
21
22    //jumping
23    public float jumpHeight = 2f;
24
25    // Update is called once per frame
26
27    // Update is called once per frame
28    void Update()
29    {
30        //keep this order
31        playerInput();
32        checkGround();
33        jump();
34        direction();
35    }
36
37    void playerInput()
38    {
39        //Player input
40        x = Input.GetAxis("Horizontal");
41        z = Input.GetAxis("Vertical");
42
43    void checkGround()
44    {
45        //checks ground
46        isGrounded = Physics.Raycast(groundCheck.position, groundDistance, groundMask);
47
48        if(isGrounded && velocity.y < 0)
49        {
50            velocity.y = -2f;
51        }
52
53    void direction()
54    {
55        //direction
56        Vector3 move = transform.right * x + transform.forward * z;
57
58        controller.Move(move * speed * Time.deltaTime);
59
60        velocity.y += gravity * Time.deltaTime;
61
62        controller.Move(velocity * Time.deltaTime);
63
64    void jump()
65    {
66        if(Input.GetButtonDown("Jump") && isGrounded)
67        {
68            velocity.y = Mathf.Sqrt(jumpHeight * -2 * gravity);
69        }
70    }
71
72    }
73
74
75
76

```

Figure 14 - playerMovement.cs



```

1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  public class mouseLook : MonoBehaviour
6  {
7
8      float mouseX = 0f;
9      float mouseY = 0f;
10
11     public float mouseSensitivity = 400f;
12
13     public Transform playerBody;
14
15     float xRotation = 0f;
16
17     // Start is called before the first frame update
18     void Start()
19     {
20         Cursor.lockState = CursorLockMode.Locked;
21         Cursor.visible = false;
22     }
23
24     // Update is called once per frame
25     void Update()
26     {
27         mouseMovement();
28         rotationCam();
29     }
30
31     void mouseMovement()
32     {
33         //Gets the mouse inputs
34         //Time.deltaTime is independent of framrate so the mouse movement is
35         //consistant no matter how many Frames are being rendered
36         mouseX = Input.GetAxis("Mouse X") * mouseSensitivity * Time.deltaTime;
37         mouseY = Input.GetAxis("Mouse Y") * mouseSensitivity * Time.deltaTime;
38     }
39
40     void rotationCam()
41     {
42         //x rotation of the player object
43         playerBody.Rotate(Vector3.up * mouseX);
44
45         //y rotation of the camera object
46         xRotation -= mouseY;
47         xRotation = Mathf.Clamp(xRotation, -90f, 90f);
48         transform.localRotation = Quaternion.Euler(xRotation, 0f, 0f);
49     }
50

```

Figure 15 - mouseLook.cs

Blender

Blender is a free 3d modeling tool used to make 3d assets for games, movies, shows, CAD, and Cam. It supports the entire 3d pipeline which includes, modeling, rigging, simulation, rendering, compositing, motion tracking, video editing and game creation (Blender Foundation, n.d.).

A tutorial that was followed during the development project to make 3d assets for the lighting demo. The first asset that was created was a low poly building (Ryan King Art, 2022). This was a huge learning curve as blender has a very confusing UI, and there weren't many beginner-friendly tutorials online. After making the low poly building, a few more assets were created for the game. Figure 17 shows a low poly house with textures. The way and techniques learnt from making the house made it easier to create other assets.

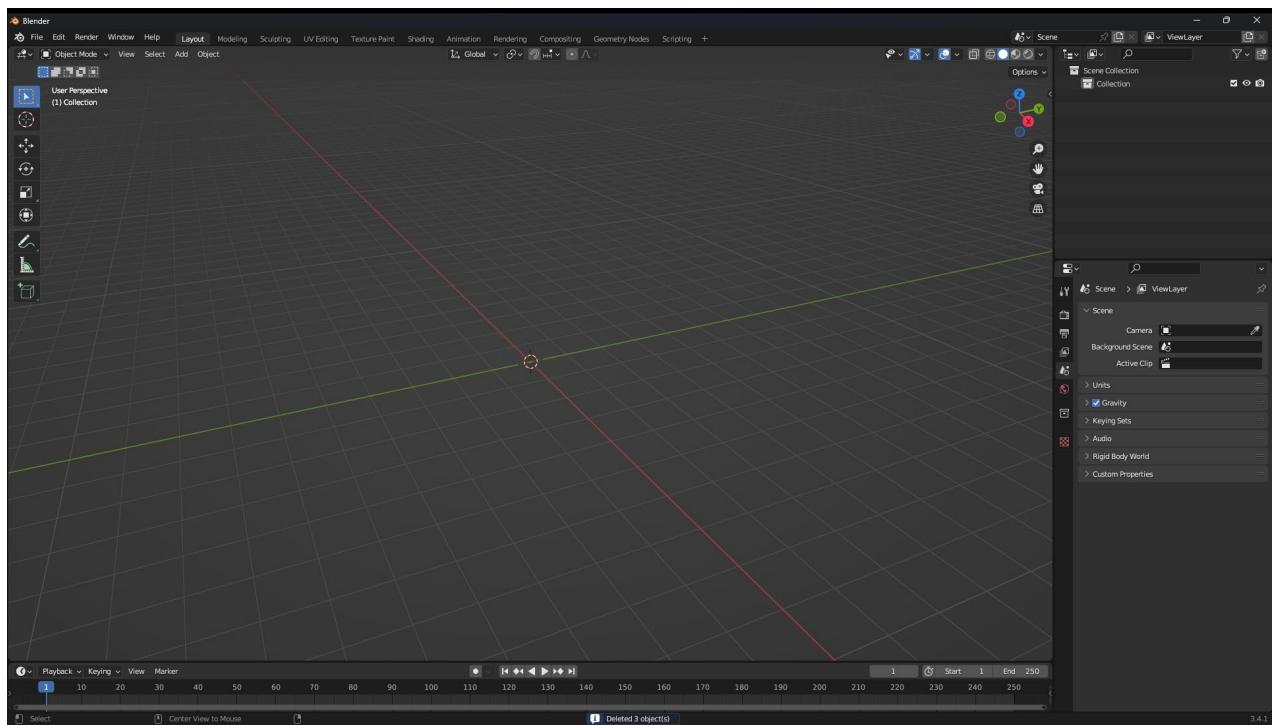


Figure 16 - Empty Blender Project

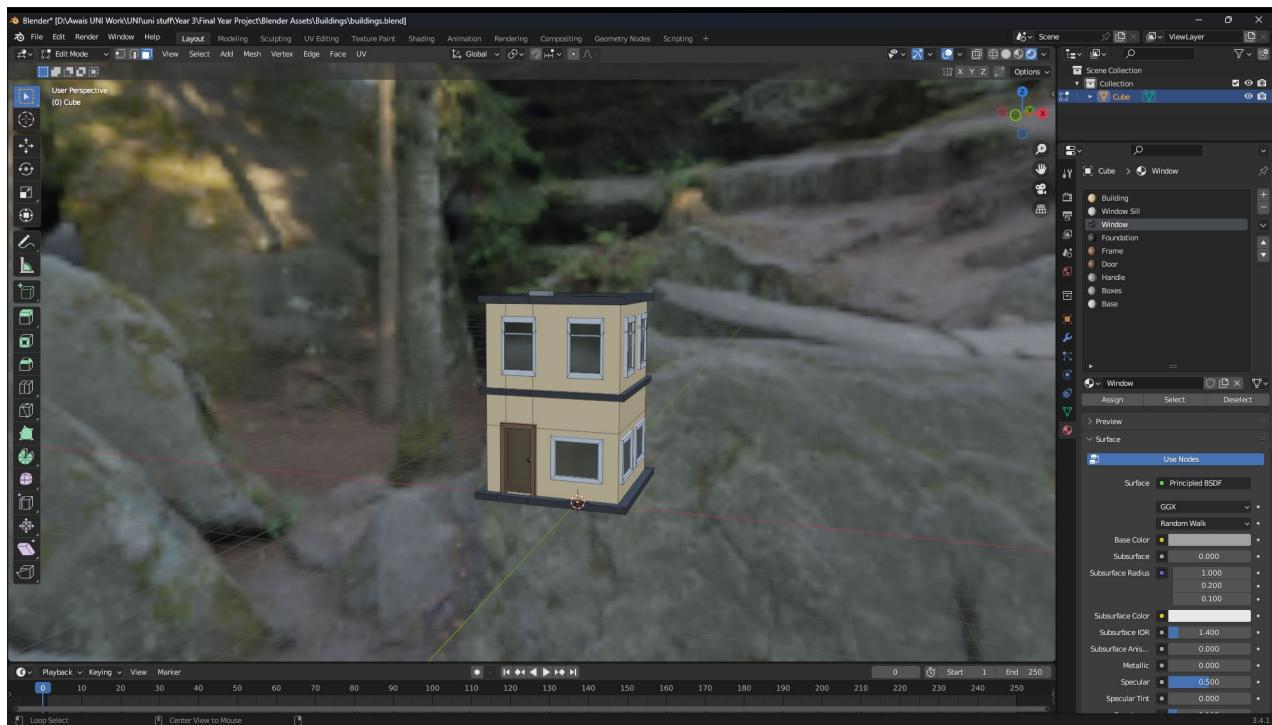


Figure 17 - Low Poly House

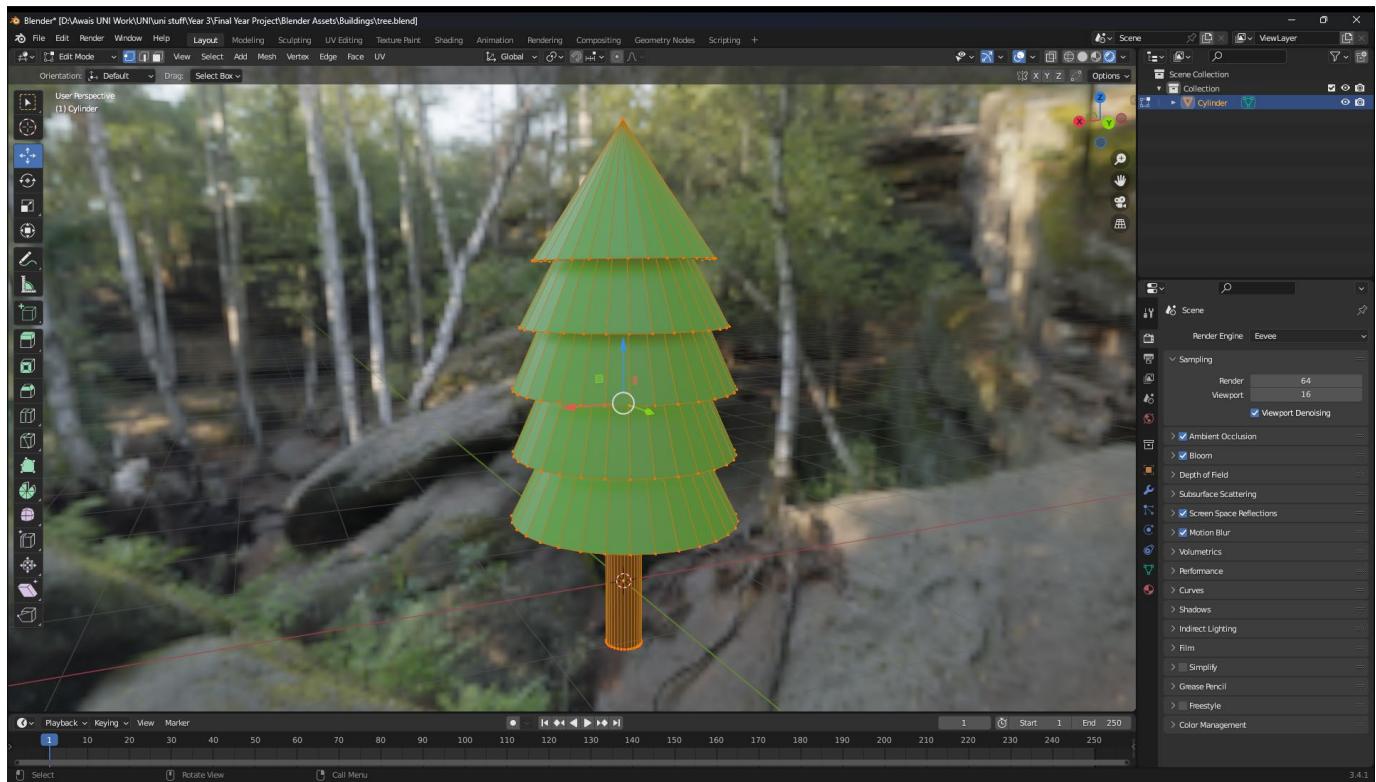


Figure 18 - Low Poly Tree

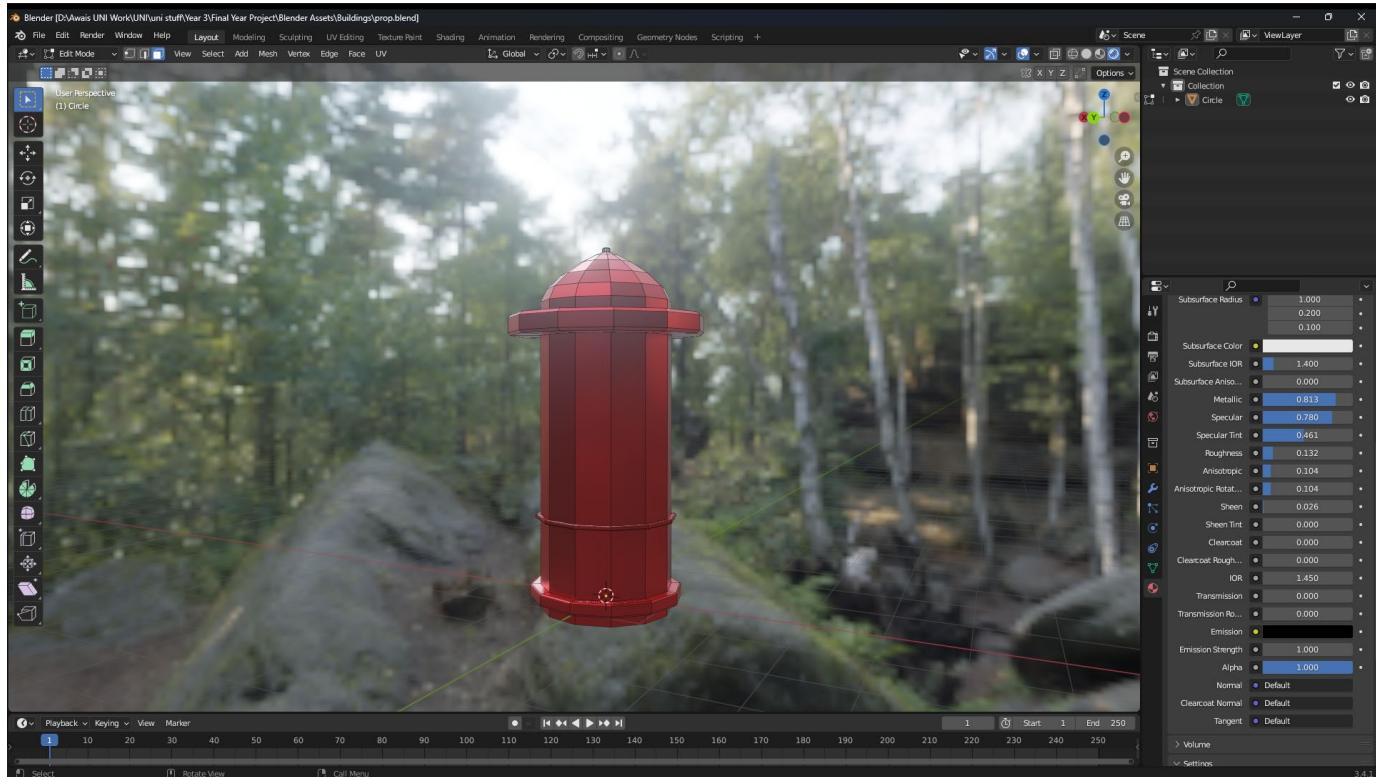


Figure 19 - Low Poly Fire Hydrant

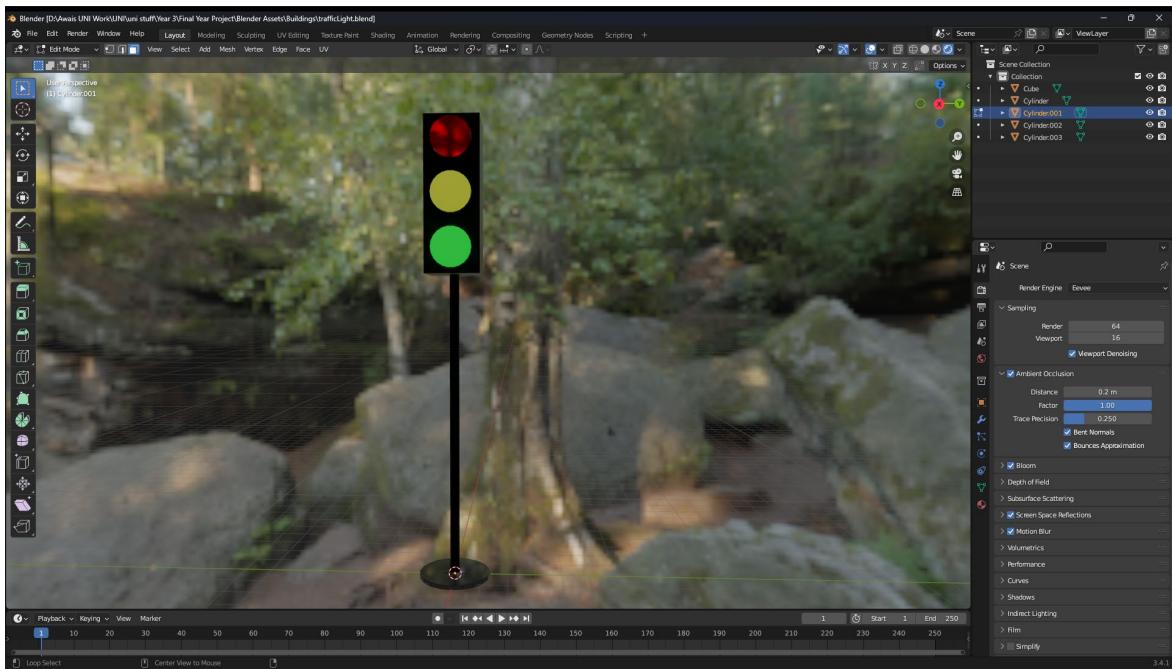


Figure 20 - Low Poly Traffic Light

World Creation

This was the first attempt at making a game world during development. The floor was a simple cube extended out both in the x and z axis. A few basic assets were added in such as stairs, walls, lights, cubes and then the blender models were added into the game world. Also, it was a good way to test the player's movement.

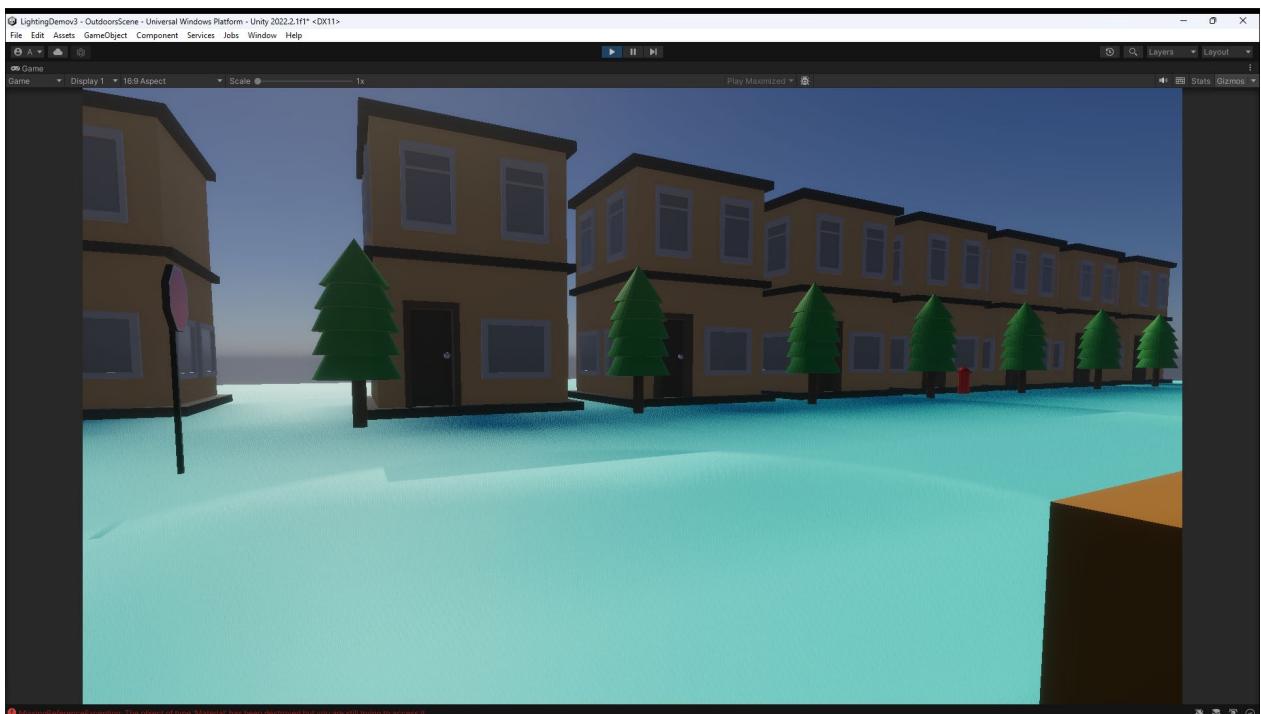


Figure 21 - Gameplay

Terrain

Terrain is the next step in good map design. The terrain was created using Unity's built in terrain maker tool, and a tutorial by Brackeys (Brakeys, 2019) was followed. The terrain shape was made by using a heightmap. A height map is a greyscale texture that stores the information on landscapes, for example the details of a hill or valley (Unreal Engine, 2020). The terrain in the game would have the exact same features as the heightmap. Also, some vegetation such as bushes and grass has been added to the terrain.

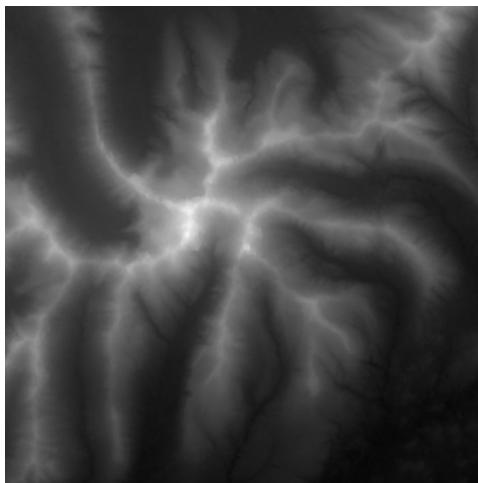


Figure 22 - Heightmap

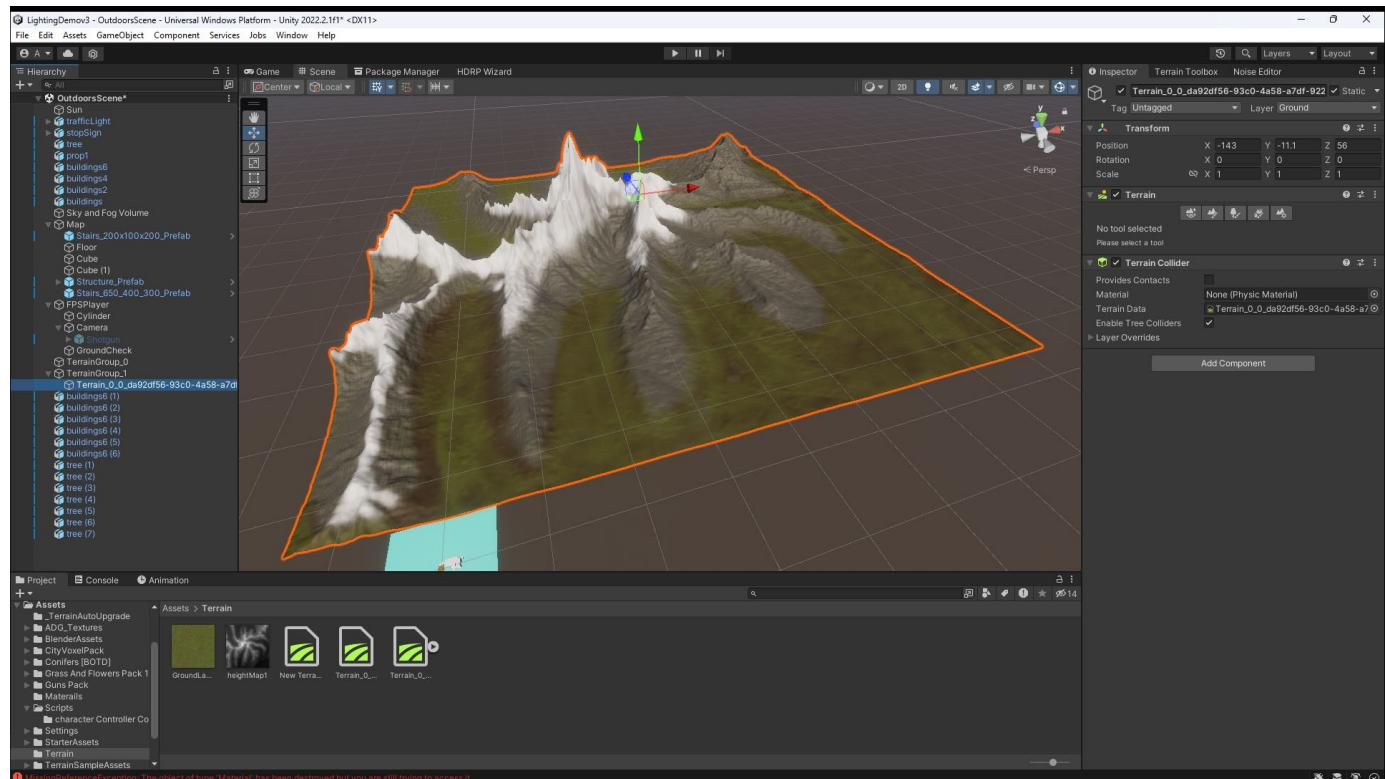


Figure 23 - Overview of Terrain

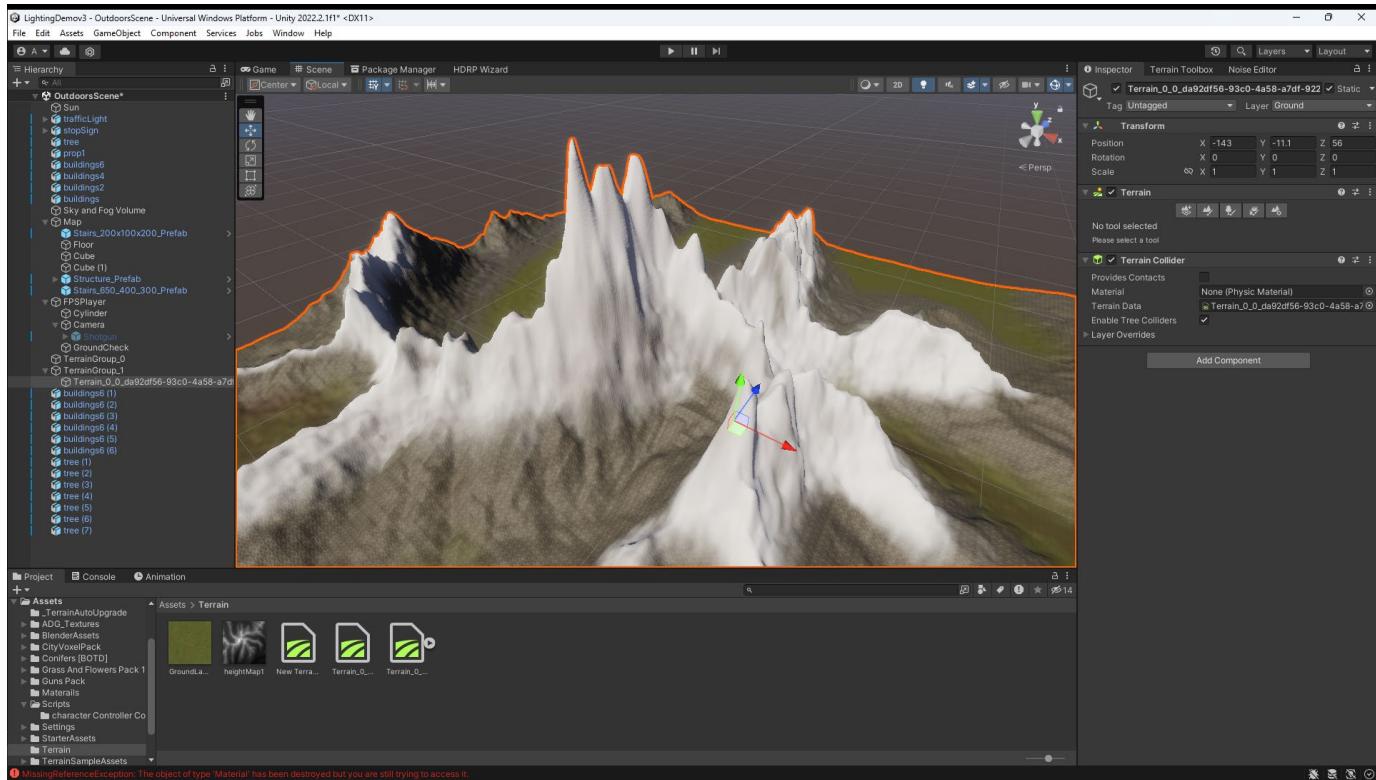


Figure 24 - Zoomed in view of Terrain.



Figure 25 - Vegetation

SPRINT 2 – FINAL RELEASE

Sprint 2 is split into 3 parts. Level one is going to show the player how baked lighting works in Unity. Level 2 is going to show the player what actual real time lighting is going to look like in Unity. Level 3 is about informing the player about the security problems in the gaming industry and how to avoid getting hacked by giving the player specific goals to achieve.

Throughout this sprint, the UI will also be made. This section will also show how the UI was developed and how it has changed from the initial wireframe designs in the previous section.

Level 1

The main aim for level one is to show the player how Unity fakes its lighting by baking the lighting data beforehand in a lightmap. When the game runs, Unity applies the lightmap to the scene geometry like a texture on a model (Unity, 2020). This saves a lot of GPU power as the light rays don't need to be calculated on the fly. This technique is like rasterized graphics where the calculations for the 3d models are done beforehand and saved to a bitmap file to reduce the load on the GPU at runtime (CAULFIELD, 2018).

The secondary aim was to show how lighting can make players make decisions, such as which path to take based on where the light is.

The first part that was worked on for this level was making the corridor for this level. The level was made in the Unity editor instead of Blender. This is to show that you can do some modeling work in Unity.

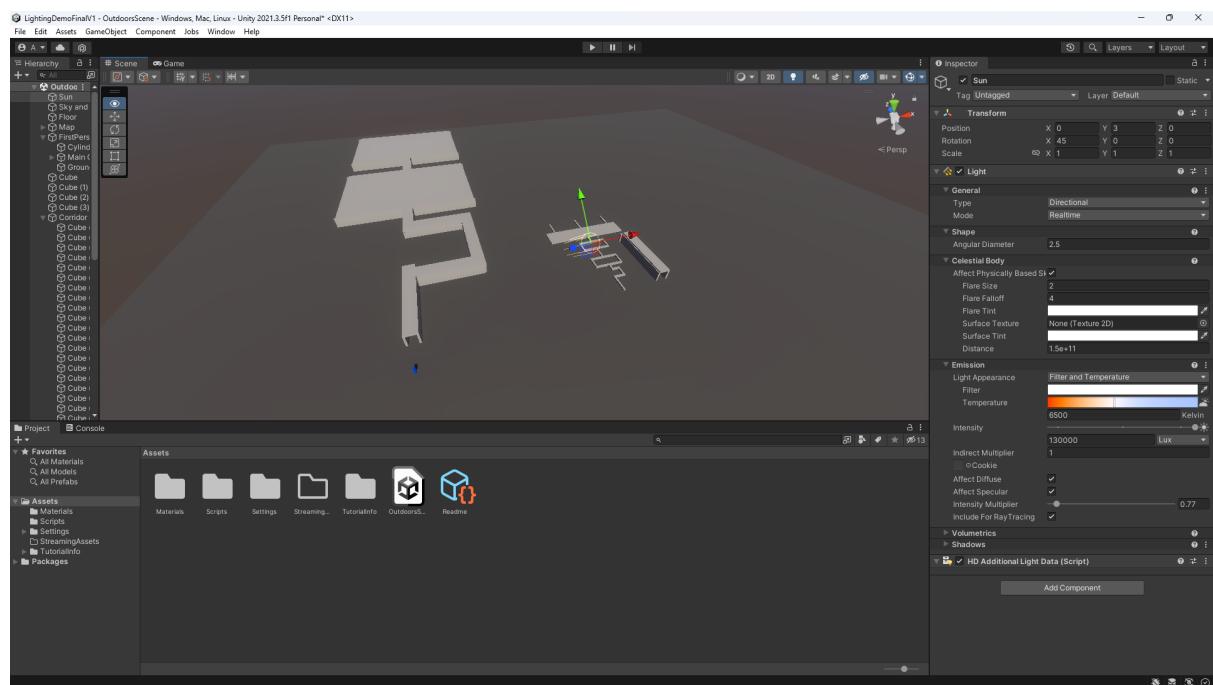


Figure 26 – Corridor

The next part of level 1 was adding the movement mechanics to the scene. The version that was in the MVP was improved to make it easier to move around and to easily lock and unlock the mouse from the game scene to make it easier to use the UI.

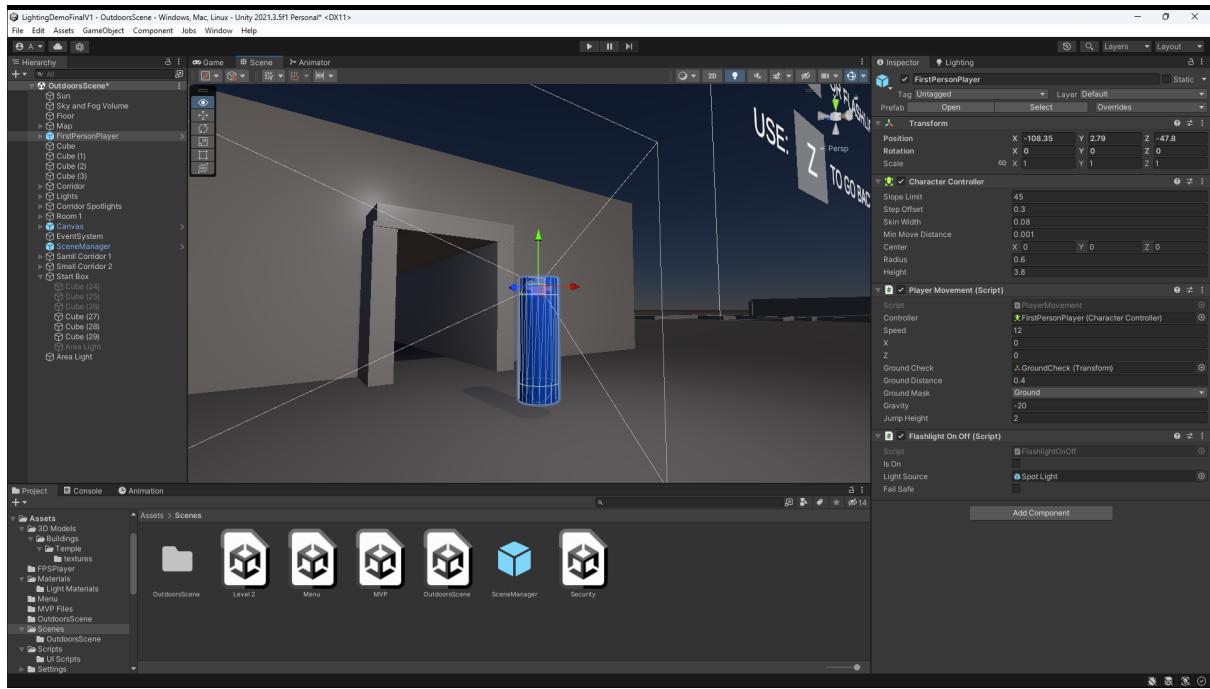


Figure 27 - Level 1 Movement Camera

```
public class LockMouse : MonoBehaviour
{
    // Start is called before the first frame update
    void Start()
    {
        //this locks the mouse to the center of the screen and also makes the mouse invisible
        Cursor.lockState = CursorLockMode.Locked;
        Cursor.visible = false;
    }
}
```

Figure 28 – LockMouse.cs

```
public class UnlockMouse : MonoBehaviour
{
    // Start is called before the first frame update
    0 references
    void Start()
    {
        //this unlocks the mouse and makes the mouse visible
        Cursor.lockState = CursorLockMode.None;
        Cursor.visible = true;
    }
}
```

Figure 29 – UnlockMouse.cs

After the movement was added, a flashlight was also added. The flashlight was added to make it easier for the player to navigate the corridor (Figure 31).

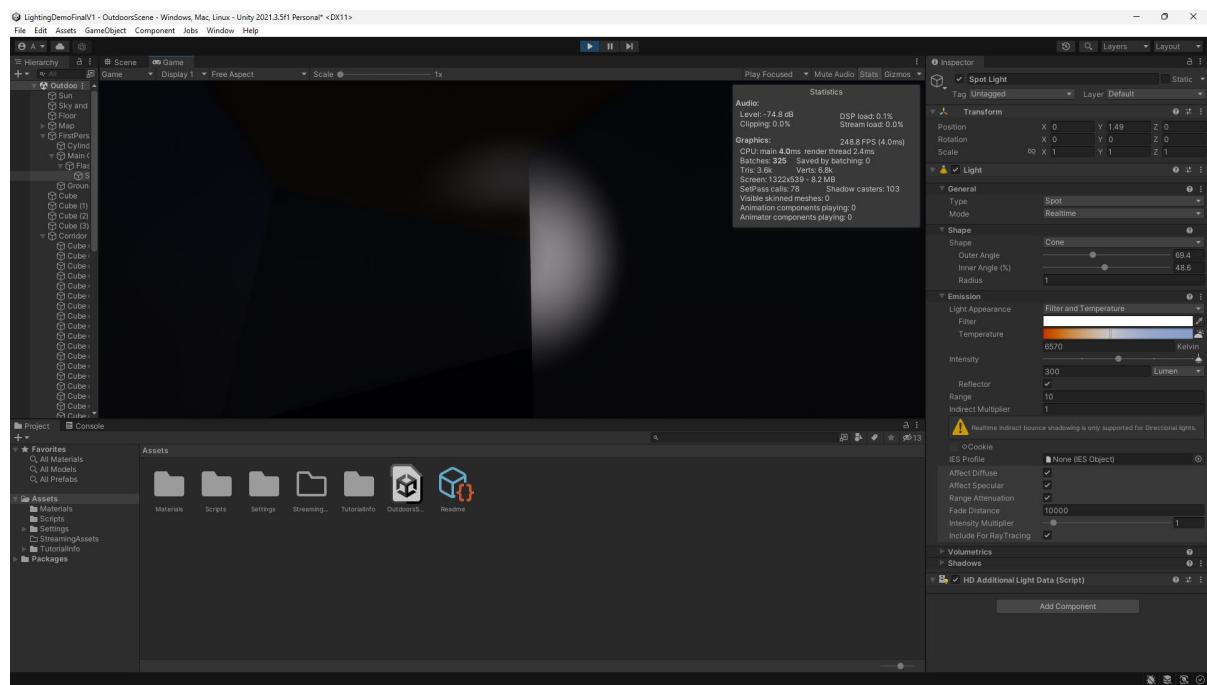


Figure 30 - Flashlight corridor

```

0 references
public class FlashlightOnOff : MonoBehaviour
{
    4 references
    public bool isOn = false;

    2 references
    public GameObject lightSource; //the flash light

    5 references
    public bool failSafe = false; //For delay

    // Update is called once per frame
    0 references
    void Update()
    {
        toggleFlashlight();
    }

    1 reference
    public void toggleFlashlight()
    {
        //This if checks to see if the F key is pressed
        if(Input.GetButtonDown("FKey")){
            //this if checks to see if the flashlight is on or off and to see if the fail safe is on
            if(isOn == false && failSafe == false){
                failSafe = true;
                lightSource.SetActive(true); //Enables the flashlight
                isOn = true;
                StartCoroutine(FailSafe()); //calls the delay
            }
            if(isOn == true && failSafe == false){
                failSafe = true;
                lightSource.SetActive(false); //turns off the flashlight
                isOn = false;
                StartCoroutine(FailSafe()); //calls the delay
            }
        }
    }

    2 references
    IEnumerator FailSafe(){
        yield return new WaitForSeconds(0.05f); //adds a delay of 0.05 seconds
        failSafe = false; //sets failsafe to false
    }
}

```

Figure 31 - FlashlightOnOff.cs

After the flashlight was added, the next thing to work on was the corridor lighting. This was added by using a few area lights and spotlights. Area lights based on a rectangular space. The light emitted in all directions but goes from one side of the rectangle to the other (Unity, 2016). Spotlights are in a point in space, and it sends light out in only one direction. Spotlights have a cone shape as they are constrained to an angle (Unity, 2016).

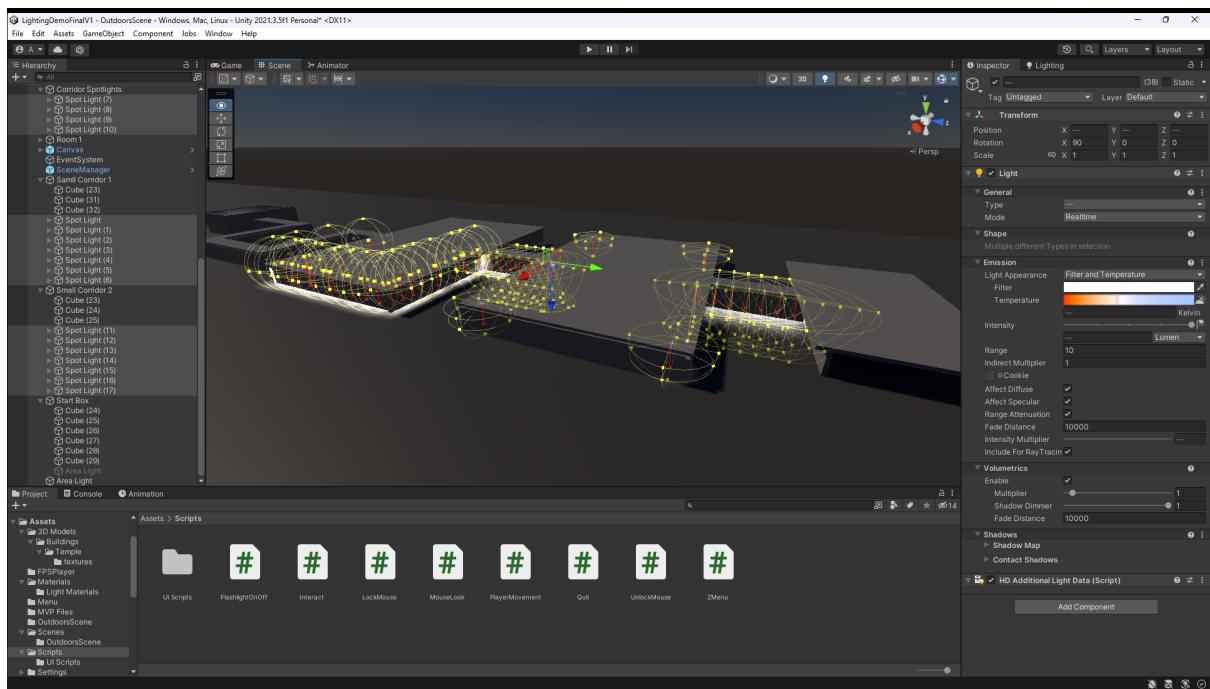


Figure 32 - Area Lights and Spotlights

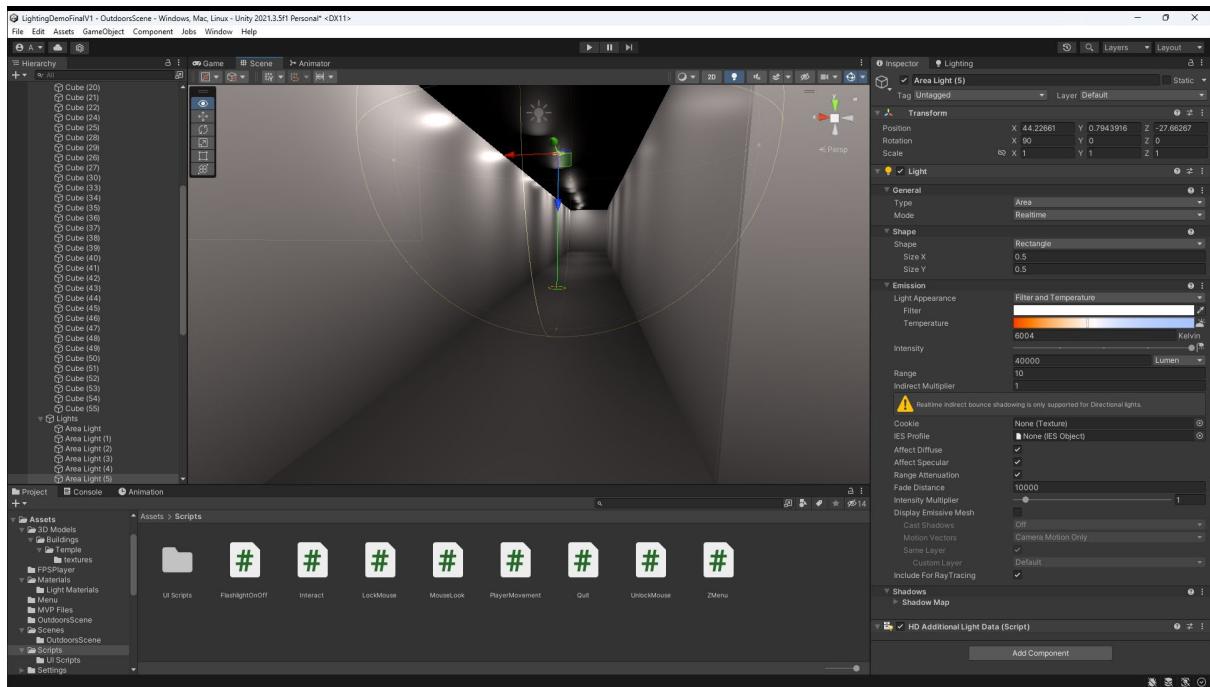


Figure 33 - Interior Area Lights

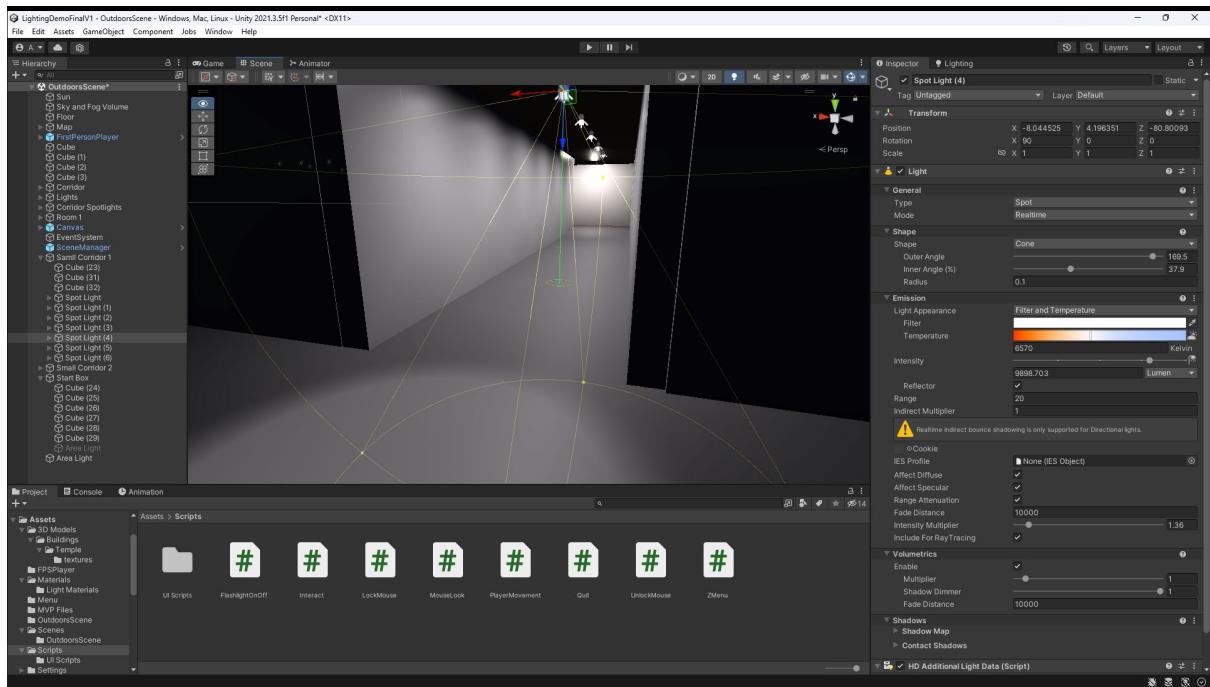


Figure 34 - Interior Spotlights

This level contains 3 decisions that the player needs to make. The light directs the player to right way. In Figure 35 is decision 1, the player needs to decide if they are going to go towards the light or towards the dark. In Figure 36 decision 2, the player has the choice to carry on walking forward into the dark, or they can walk to the right and go into the light. In Figure 37 decision 3, the player has 2 similar options, both the corridors have a light at the end of the corridor, this will make the player think about the best option.

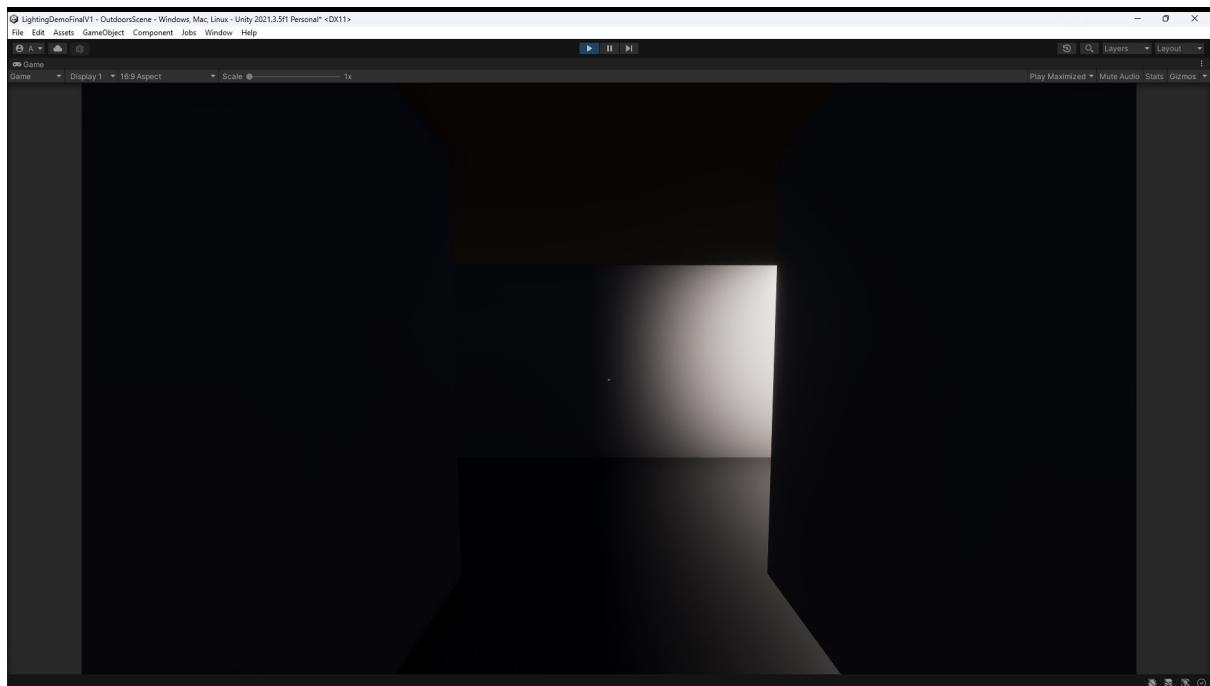


Figure 35 - Decision 1

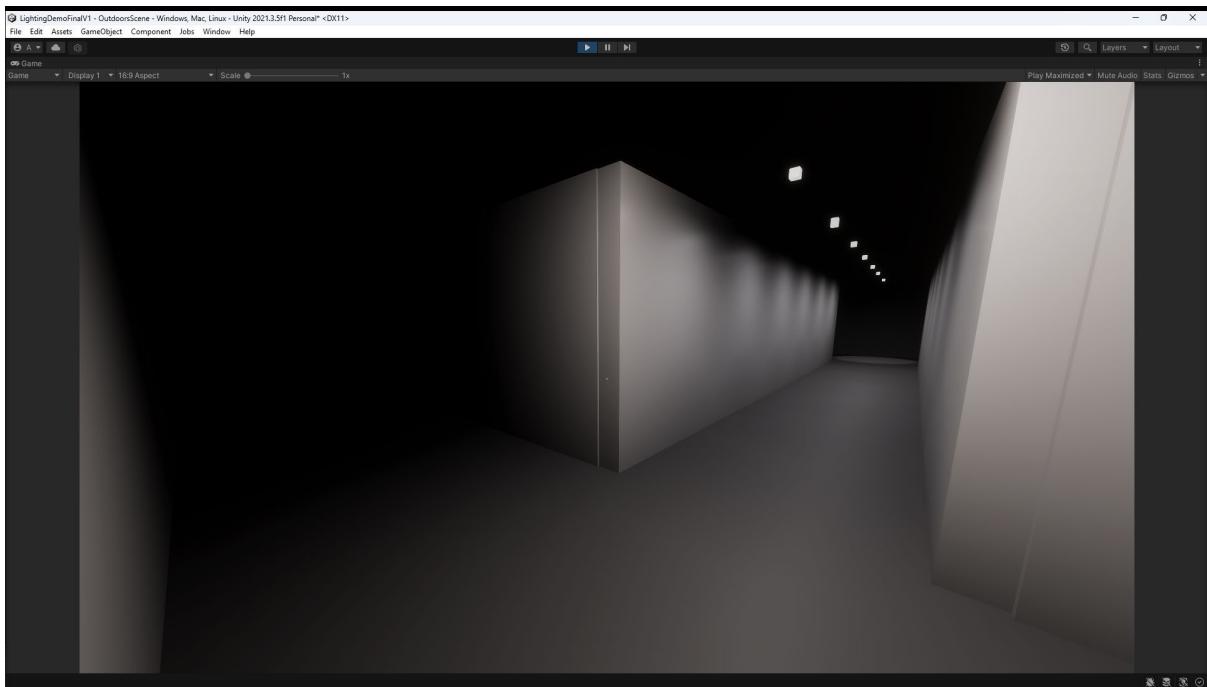


Figure 36 - Decision 2

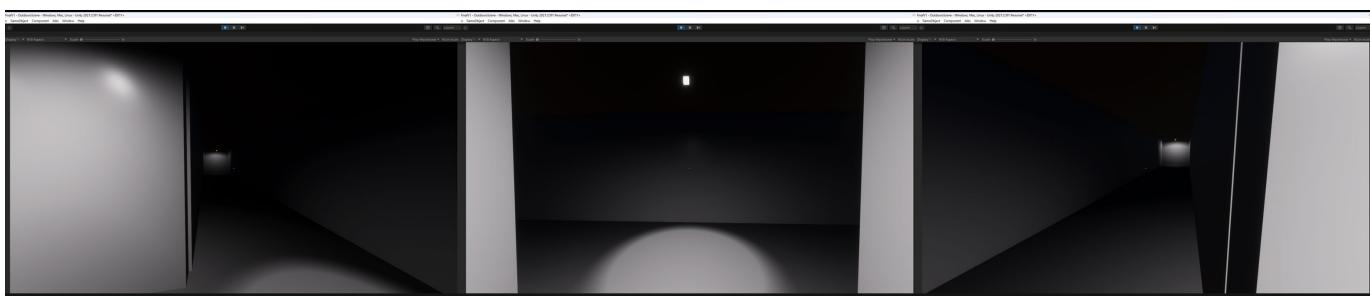


Figure 37 - Decision 3

Finally, the last part of level one is the lighting room. The lighting in this room is all baked before runtime to increase the frame rate runtime, whilst also making the lighting in the game look amazing and detailed.

The first part was to make the room and fill it with objects. These objects were downloaded from online 3d model websites as this was allowed to happen. These are a monkey head (Victor, 2014), low poly rocks (selamcukulatam, 2021), office chair (theflyingtim, 2017), generic rocks (StudioQuadro, 2021), white round table (edson-lopes, 2015).

The lighting in this room is also fully baked, and it is using emissive materials. Emissive materials emit light across the surface area of the object. For this room there are 3 rectangles that emit 3 different colored lights across all the objects in the room see Figure 38 and Figure 39. Once the room was finished and the lights were in the correct place, then Unity needed to generate the lightmap for the room by baking the lighting, see Figure 41. This process took around 2 minutes to complete (Figure 40).

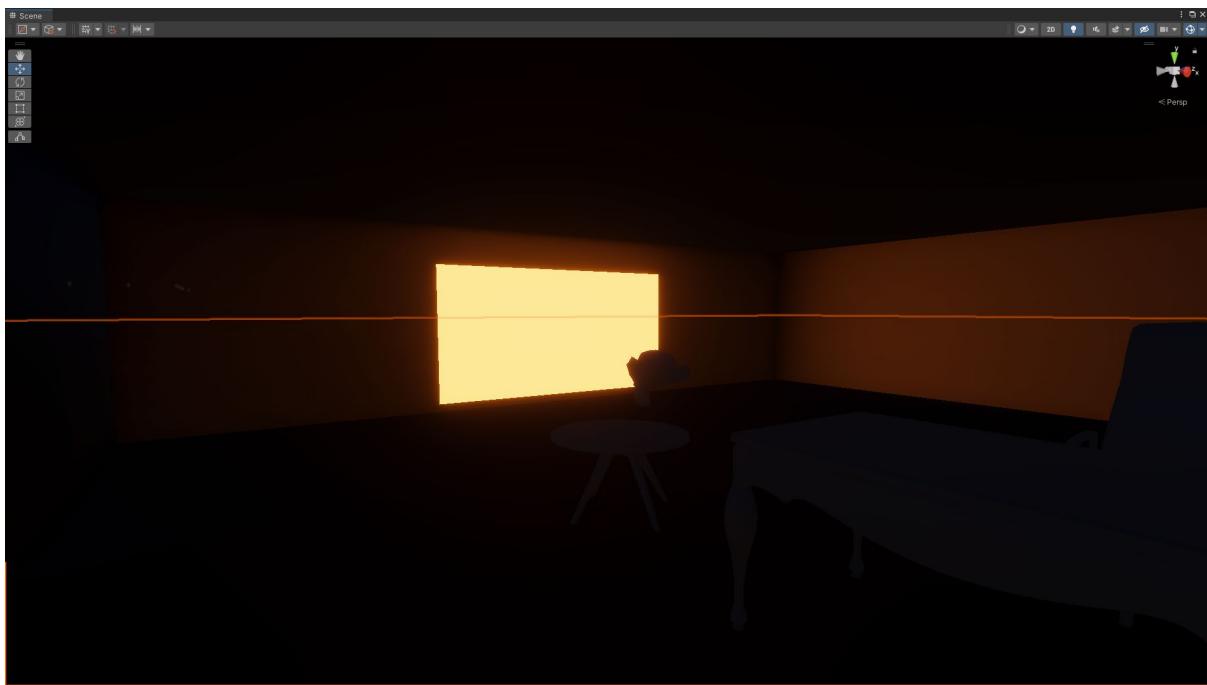


Figure 38 - Before Baking Orange Light

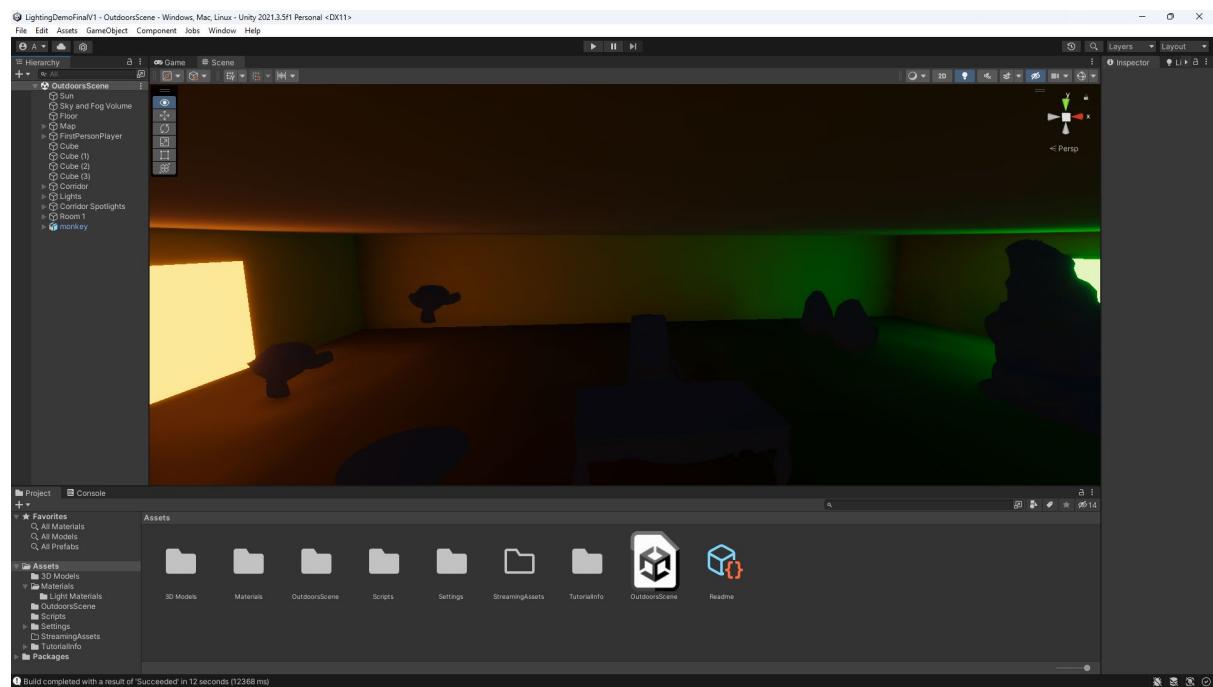


Figure 39 - Before Baking orange and Green

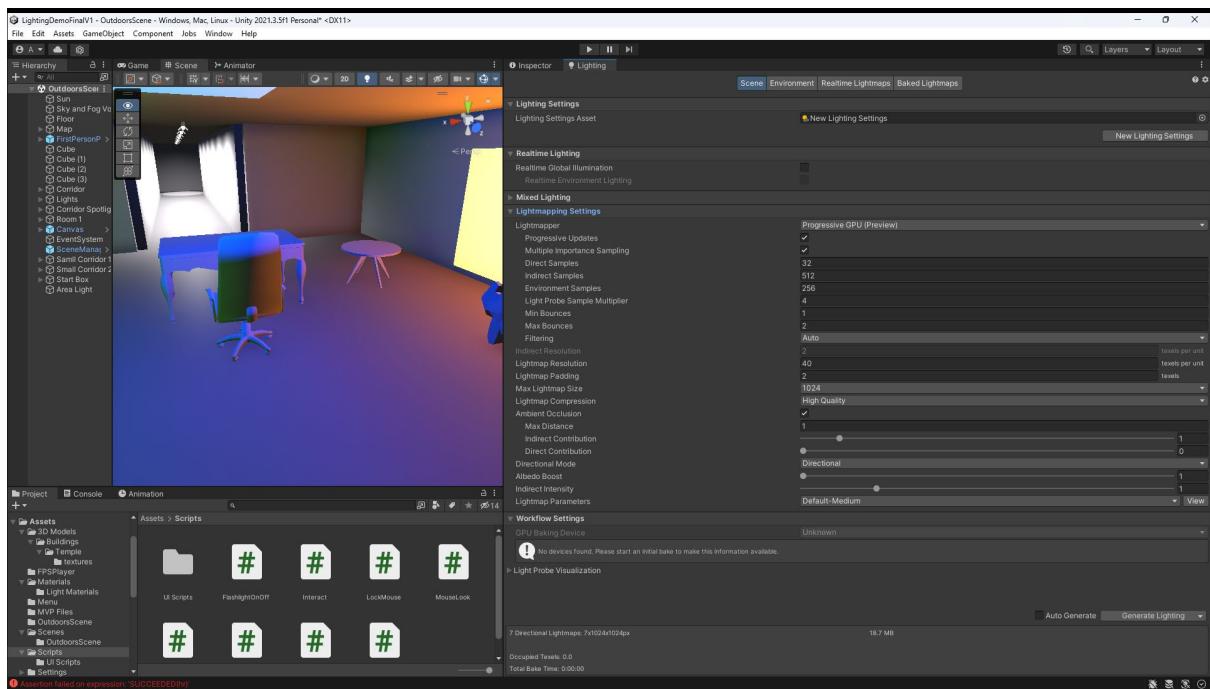


Figure 40 - Baking CPU and GPU

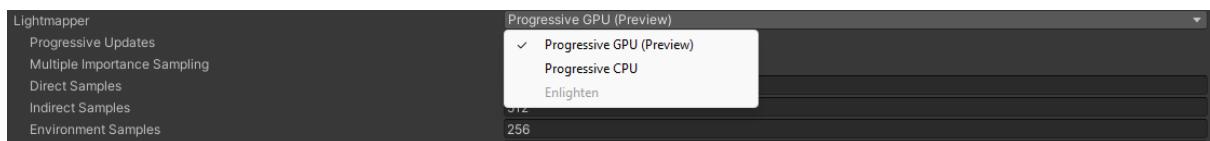


Figure 41 - Lighting Settings

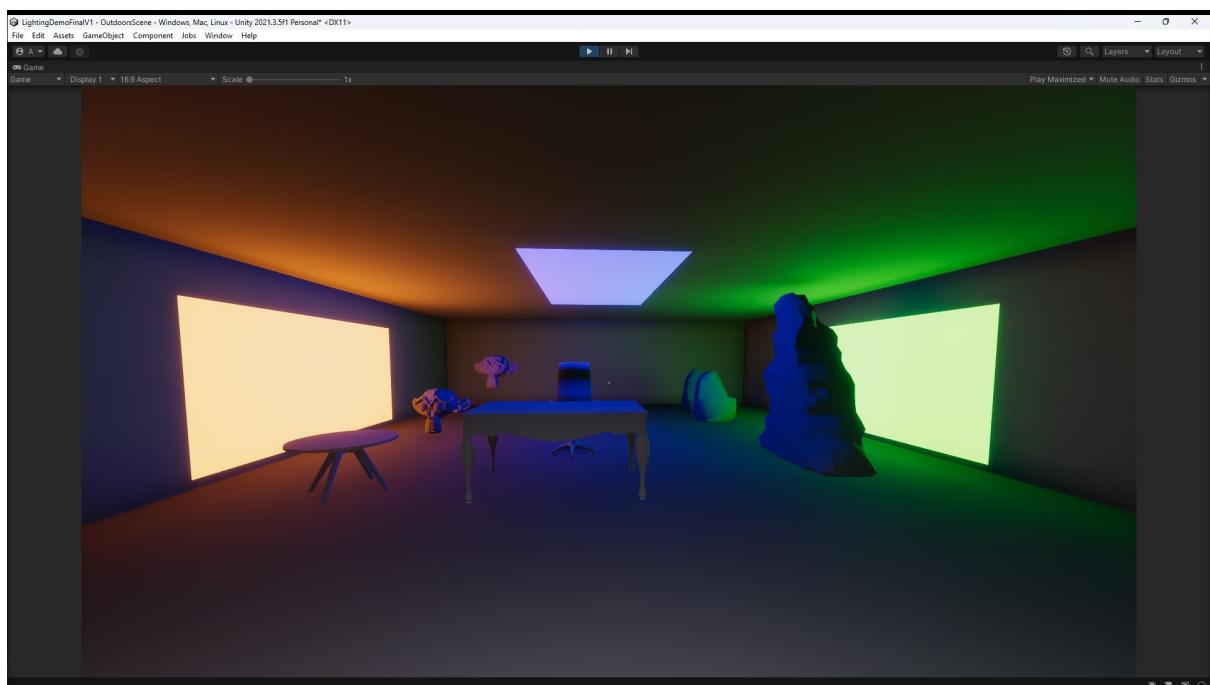


Figure 42 - After Baking with All lights

Level 2

This level's main is to show the player how Unity handles real time lighting in the Unity Game engine by having the player go around a small map based on a Mayan temple. This level also has a sun, which is a directional light. Directional light is a light that creates the effect of having a sun in the scene. The light will come from the same direction anywhere in the scene. This light does not diminish, unlike other lights such as area lights and spotlights (Unity, 2016).

The models used for this level, were downloaded from (BitGem, 2017). The downloaded files had a lot of smaller models such as walls, floor, arch see Figure 44. This meant that the map could be made to fit the project.

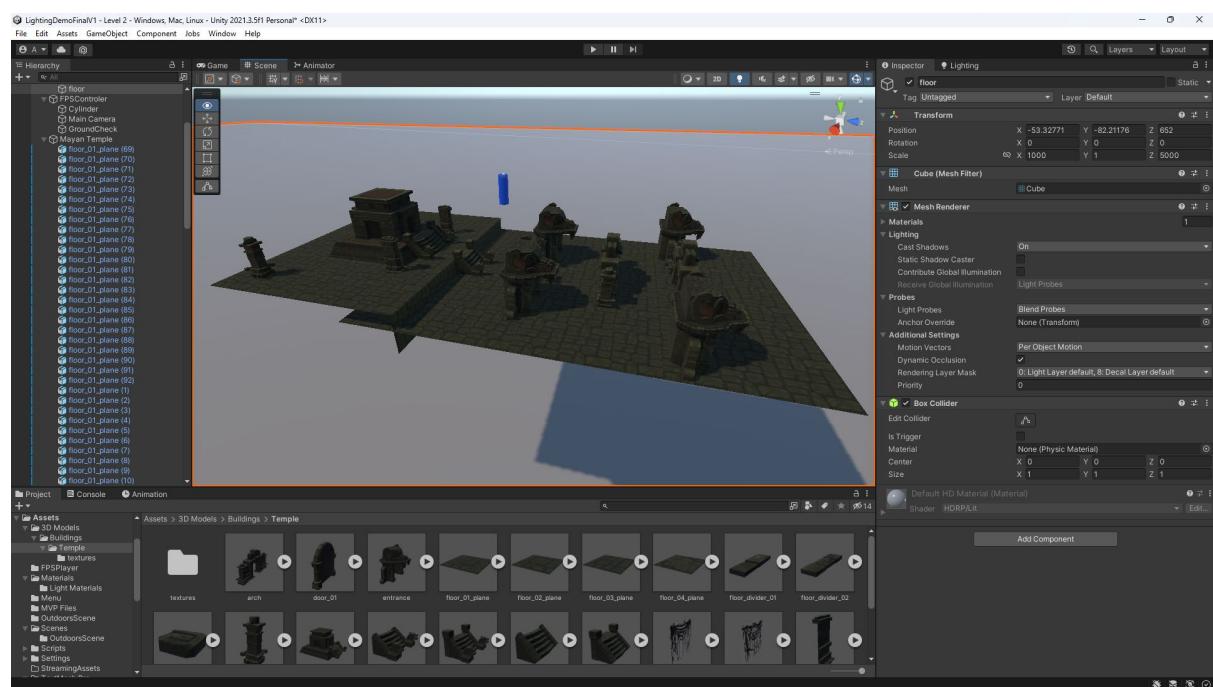


Figure 43 - Level 2 Map

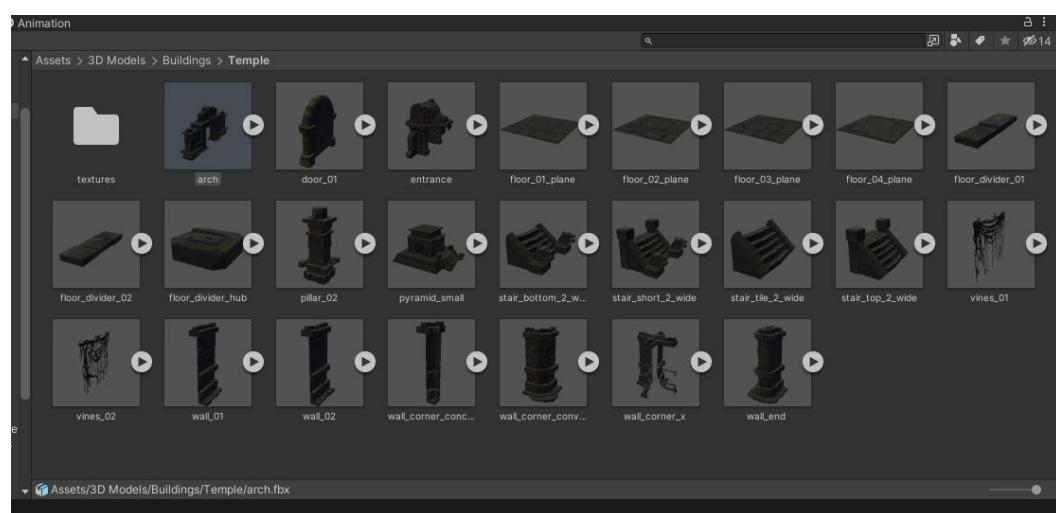


Figure 44 – Model Parts

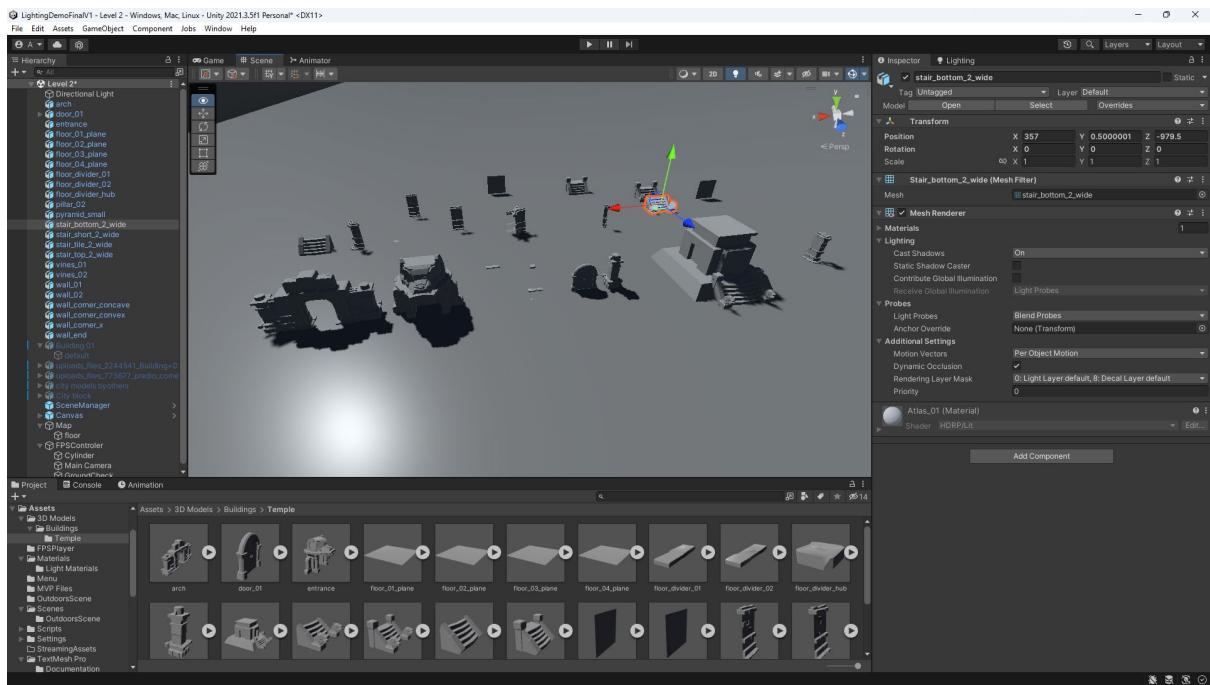


Figure 45 - Parts in Scene

Once the map was made, the player movement needed to be added, this is the same process as in Level 1.

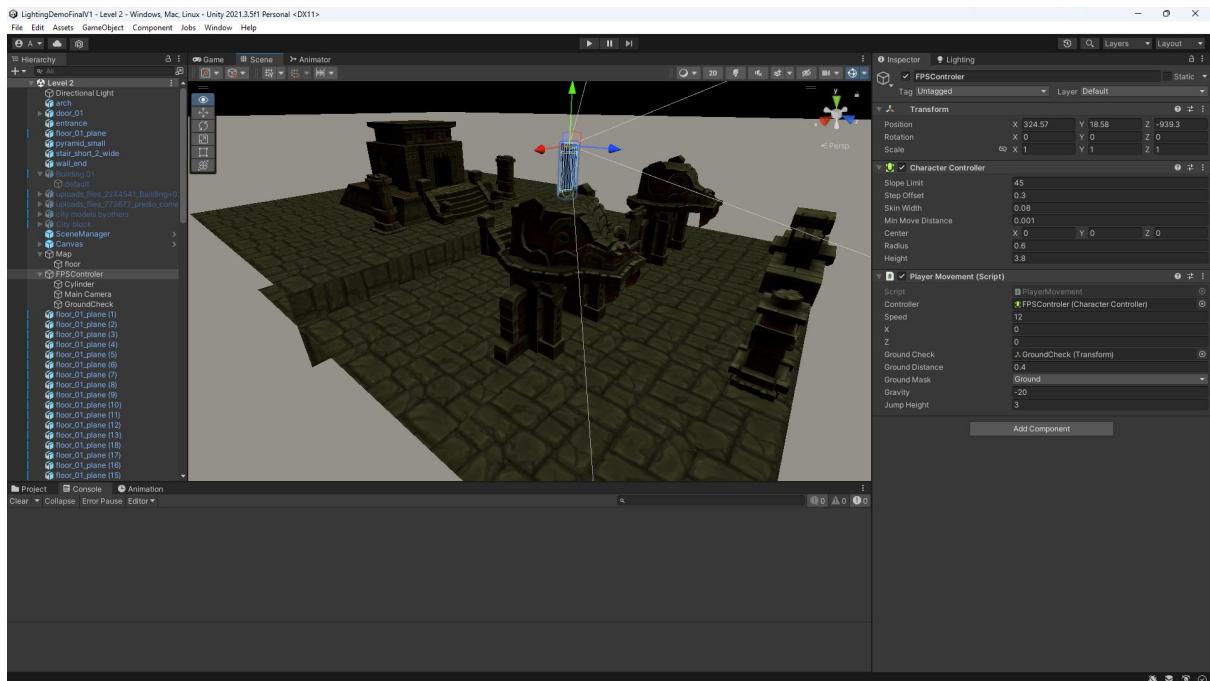


Figure 46 - Level 2 Player

Then the next step was to add lighting to the map. A light for the main temple was implemented, this light is a spotlight (Figure 47). The light emphasizes that there is something here and the player should look around. After this the light for the snakeheads were added. The 4 snakeheads have 2 separate colors, red and green (Figure 48).

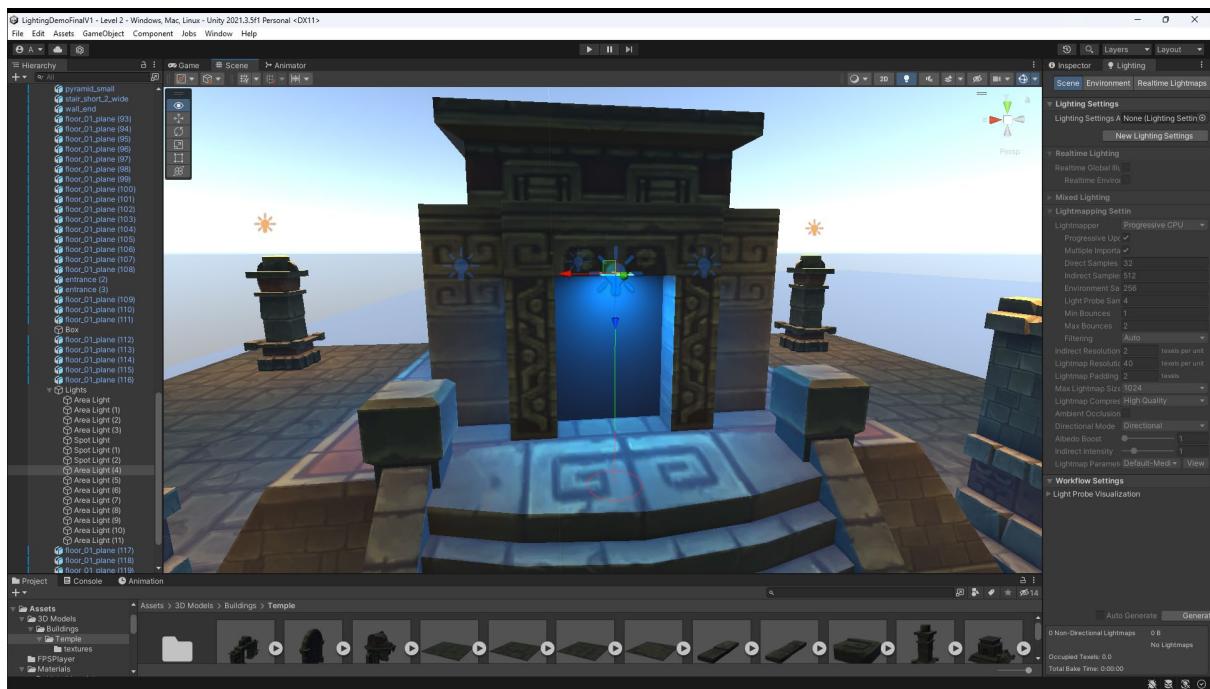


Figure 47 - Temple Lighting

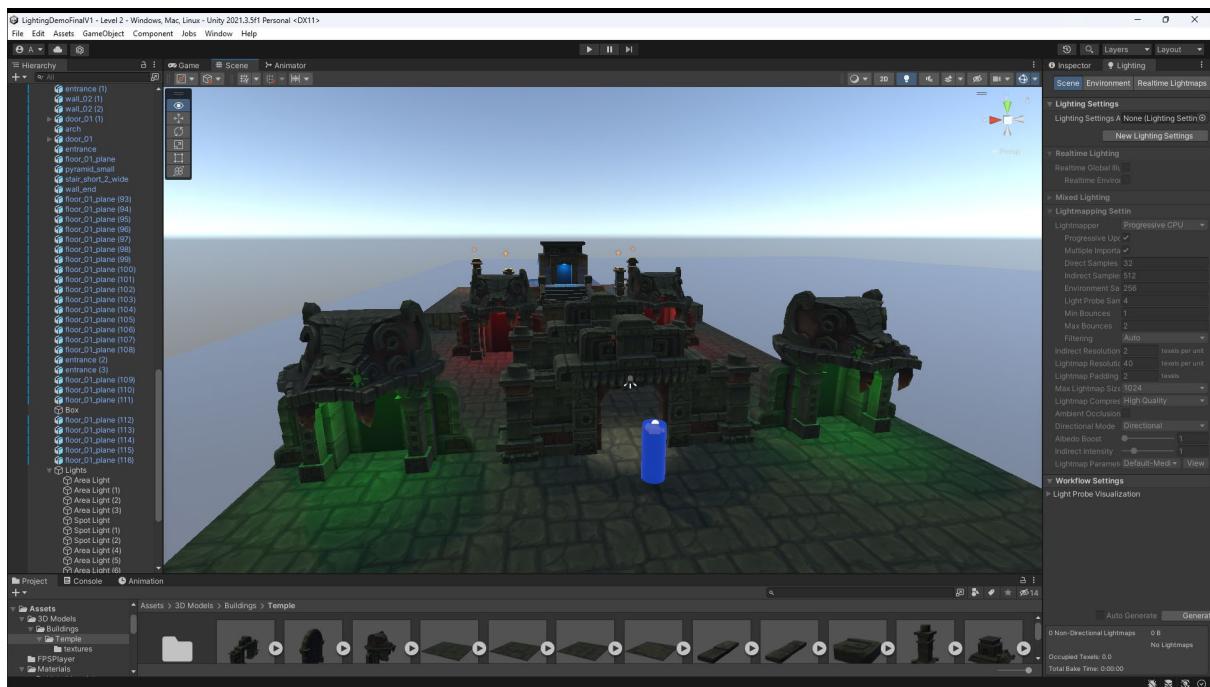


Figure 48 – Snakeheads

After the snakeheads were done, the lighting to emphasize the door and doorway was done next. This was done by using a spotlight that points at the door to attract the player's attention so that the player walks towards it.

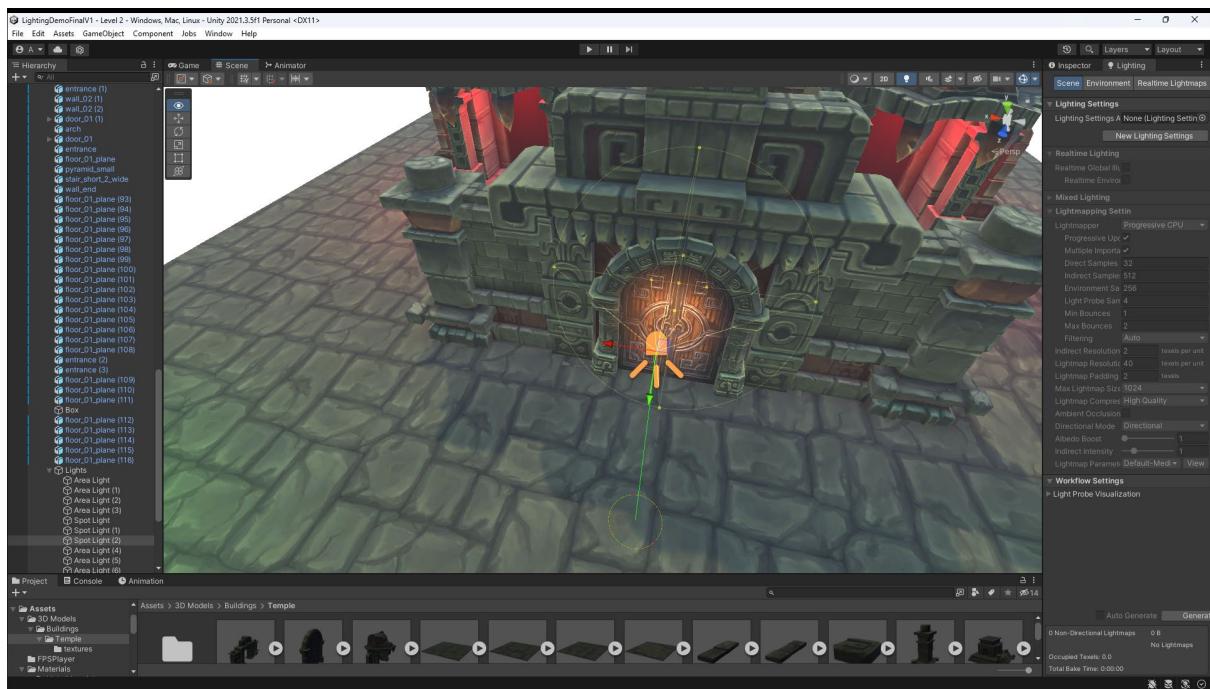


Figure 49 - Door Spotlight

After this area lights were added to pillars to mimic a fire torch on the pillar as a light source.

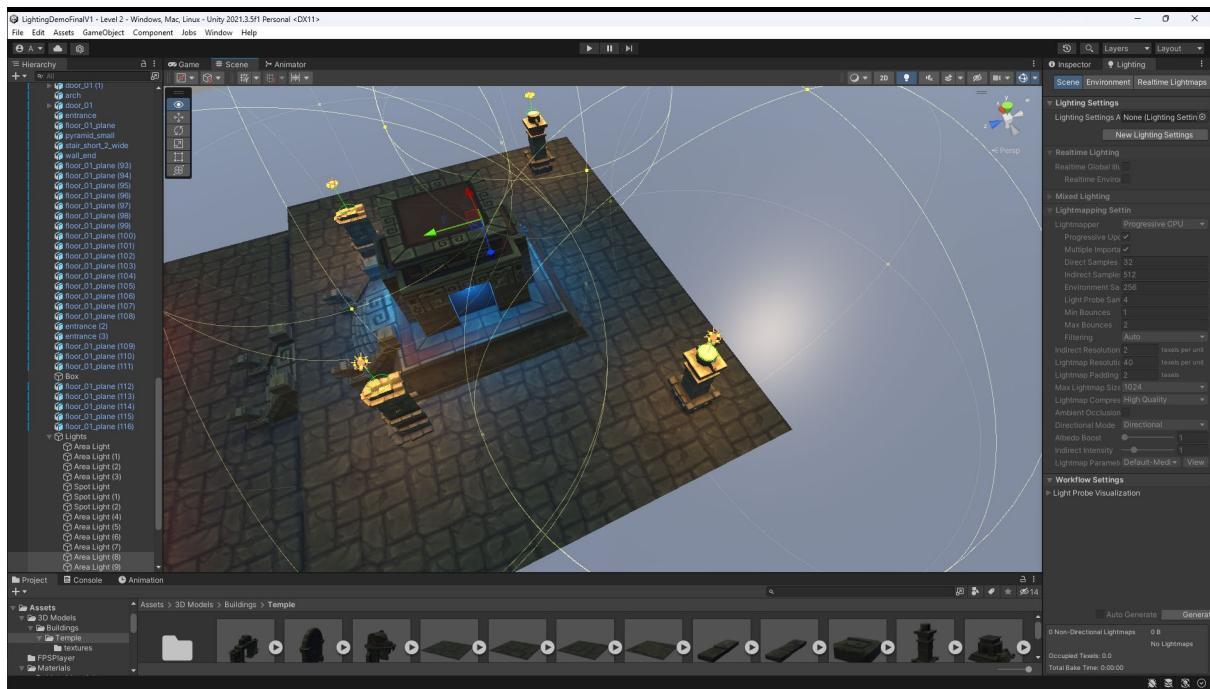


Figure 50 - Pillar Area Lights

The result of this level seen in Figure 51 shows all the lights that are in the scene.

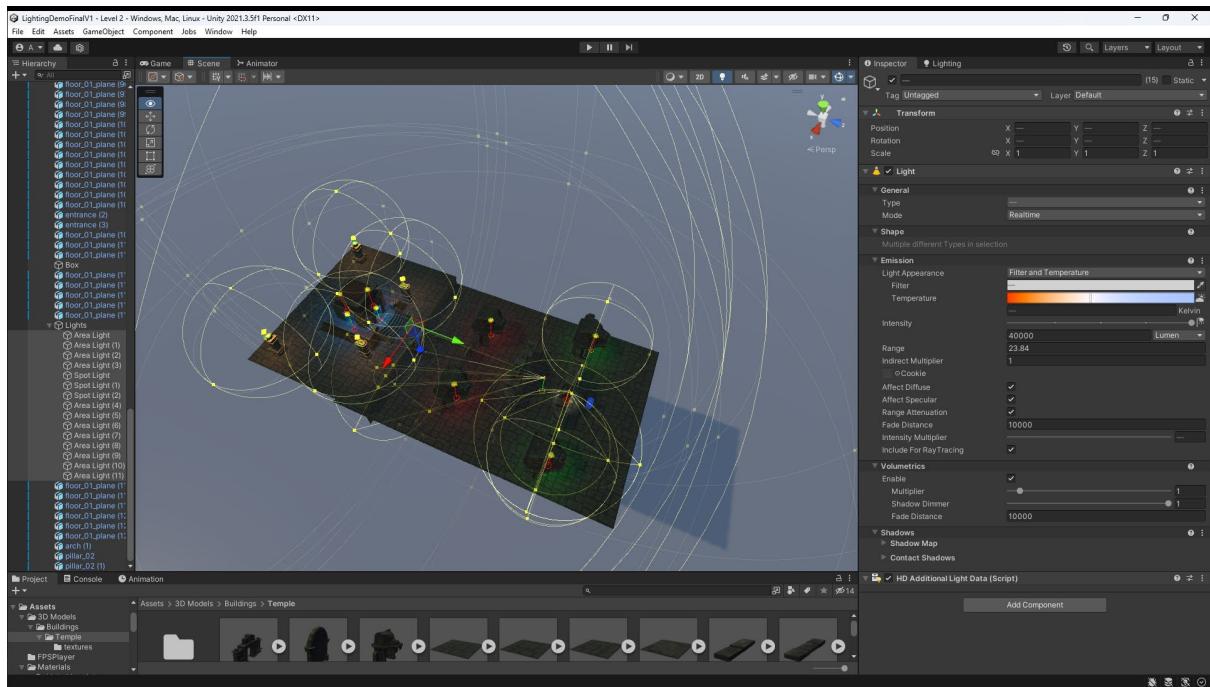


Figure 51 - Level 2 Lighting and Settings

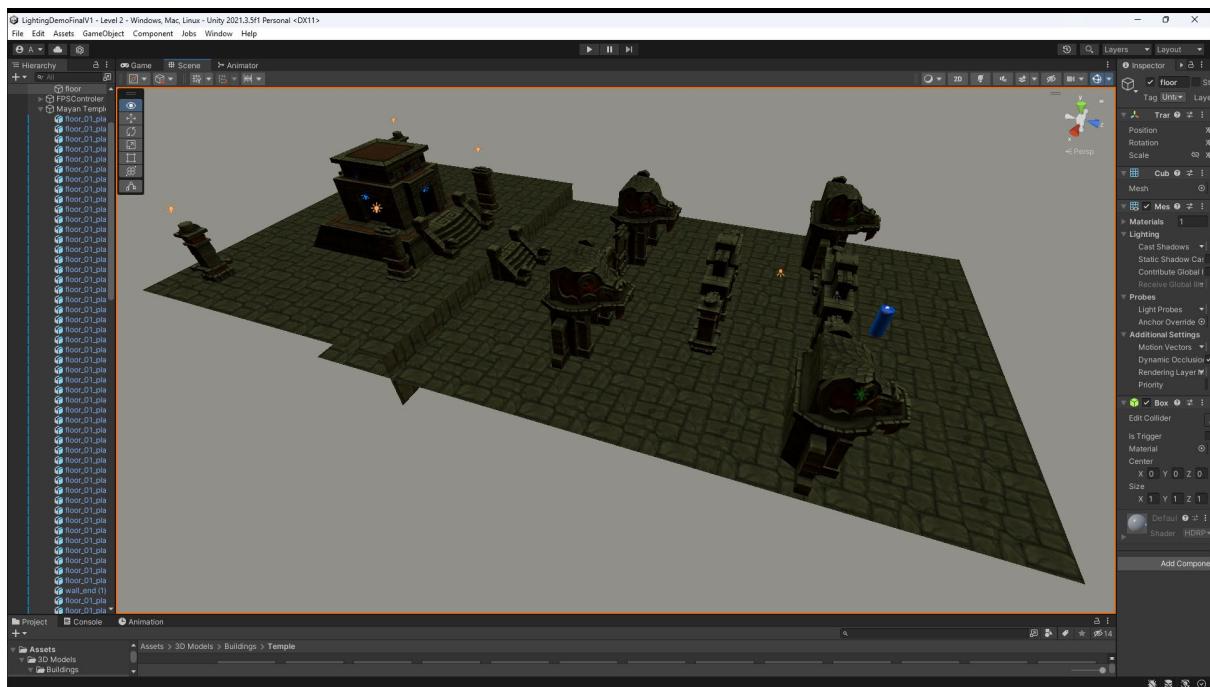


Figure 52 - Before Lighting

Figure 52 shows what the level looked like before the lighting has been added. The map looks dull and boring.

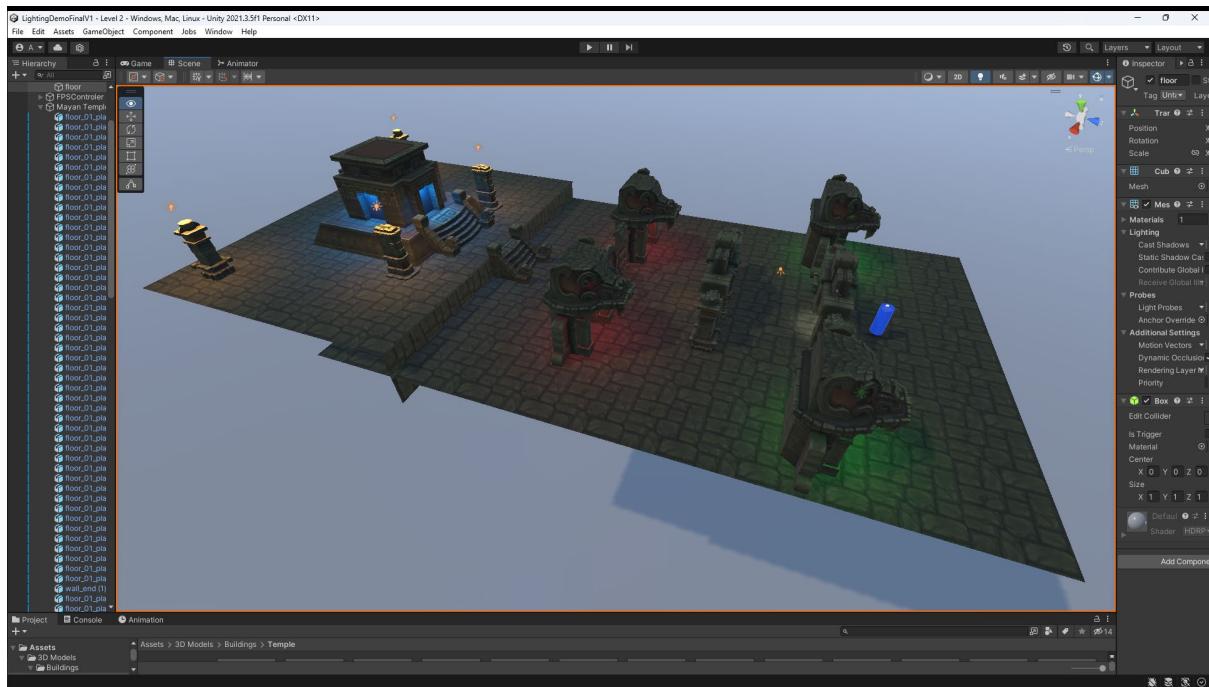


Figure 53 - After Lighting

Figure 53 shows what the map looks like after all the lighting effects have been added. This level now looks more vibrant and fuller. The other colors make the map pop more to the player. The lighting adds depth to the map by also showing shadows of the models.

Level 3

The main aim for this level is to inform the player about the different security issues in the gaming industry and it also gives the player missions on how to improve their own security.

The design of the models and graphics for this level was based on the signposts in Minecraft (Appendix h:), and the UI design and style was based on the dossiers in Call of Duty: Black Ops Cold War (Appendix i:).

Firstly, I got research on 4 security stories in the gaming industry.

GTA 5 Online hackers

Hackers hacked GTA online to get a mod menu installed to get free in game currency, which made Rockstar lose money. These hackers were also able to get other gamers IP addresses, which allowed them to carry out DDOS attacks, or steal their data (SomeOrdinaryGamers, 2023).

Call of Duty: Black Ops 3 Remote Access Trojans

Hackers were installing Remote access trojans (RAT) on to gamers machines and then they were remotely stealing the gamers data and installing key loggers and viruses on the gamers machine. The hackers were also selling these hacks to other people to use on the dark web (Naiya, 2023).

CD Projekt Red Gaming Source Code Leak

CD Projekt Red's games source code was leaked. A hacker got into CD Projekt Red's company servers and stole quite a lot of source code. One project that they stole was the source code for Cyberpunk 2077. They were later selling the stolen data for \$7 million buy-it-now price and a bidding price started at \$1 million (Porter, 2021).

Sony PlayStation Live Hack

Approximately 77 million Sony's PlayStation live network users' data was hacked and stolen by hackers. This cost Sony \$171 million. Sony said that an external intrusion had occurred, but no specifics were given. Player's passwords, date of birth, and credit/debit card were leaked (Philips, 2016).

After the research and information snippets were created, the next part to work on was the game map. This level was going to be simpler than the other to make getting the information to the player a lot easier. There are going to be 4 rooms, with 4 alters/podiums (Figure 54). As the player walks near an altar, they can press the E button (Figure 55) to bring up the dossier with the security story (Figure 57). There is also going to be a spotlight with 4 different colors to direct the players' attention to each altar. The movement is the same as the other levels.

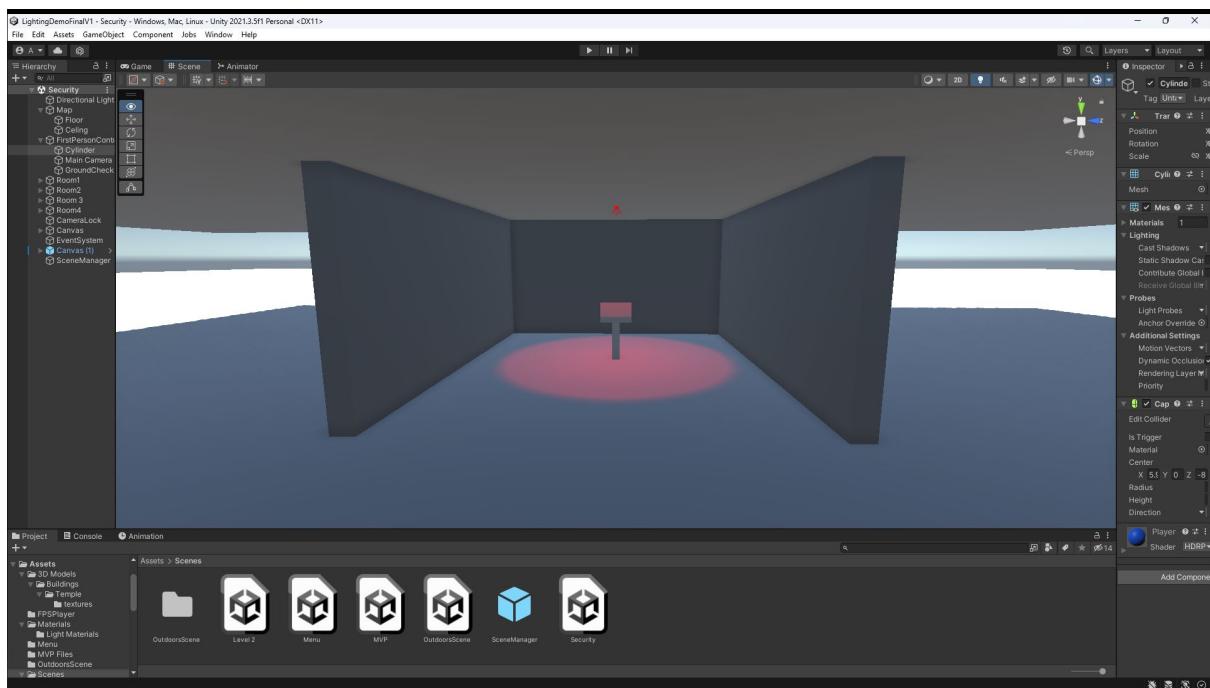


Figure 54 - Security Room

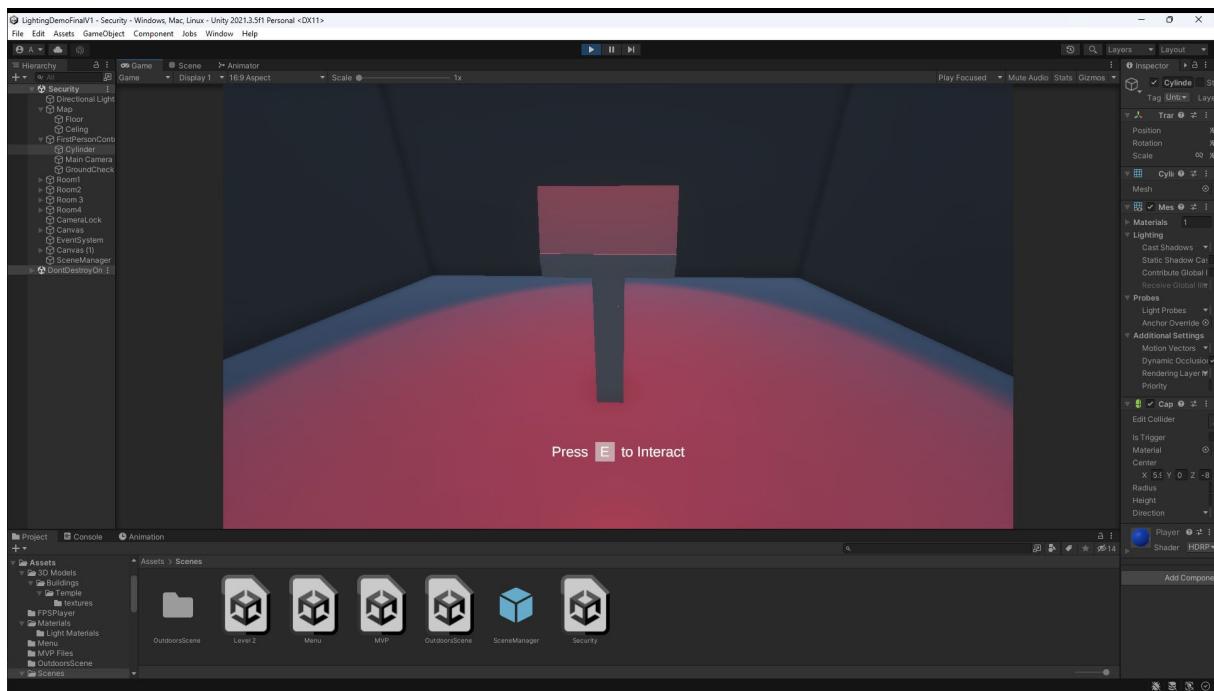


Figure 55 - E Interaction

```

1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4  using UnityEngine.UI; //Calling UI Libraries
5  using TMPro;
6
7  public class Interact : MonoBehaviour
8  {
9      public GameObject InteractUI; //the sphere around the alter
10     public GameObject ThingToPopUp; //the E to pop up
11
12     public bool failsafe = false;
13     public bool isOn = false;
14
15     //Trigger to see if player is in the sphere of InteractUI
16     public void OnTriggerEnter(Collider other)
17     {
18         InteractUI.SetActive(true); //shows the UI Element E
19     }
20
21     //Trigger to see if player is in the sphere of InteractUI
22     public void OnTriggerExit(Collider other)
23     {
24         InteractUI.SetActive(false); //Hides the UI element E
25     }
26
27     public void Update()
28     {
29         activateUI(); //Calls activateUI
30     }
31
32     //This if checks to see if the E key is pressed
33     if(Input.GetKeyDown(KeyCode.E) && InteractUI.activeInHierarchy)
34     {
35         if(isOn == false && failsafe == false)
36         {
37             failsafe = true;
38             InteractUI.SetActive(false); //sets the state of InteractUI
39             ThingToPopUp.SetActive(true); //sets the state of ThingToPopUp
40             isOn = true;
41             StartCoroutine(Failsafe()); //calls the delay
42         }
43         if(isOn == true && failsafe == false)
44         {
45             failsafe = true;
46             InteractUI.SetActive(true); //sets the state of InteractUI
47             ThingToPopUp.SetActive(false); //sets the state of ThingToPopUp
48             isOn = false;
49             StartCoroutine(Failsafe()); //calls the delay
50         }
51     }
52     IEnumerator Failsafe()
53     {
54         yield return new WaitForSeconds(0.05f); //adds a delay of 0.05 seconds
55         failsafe = false; //sets failsafe to false
56     }
57 }

```

Figure 56 - interact.cs

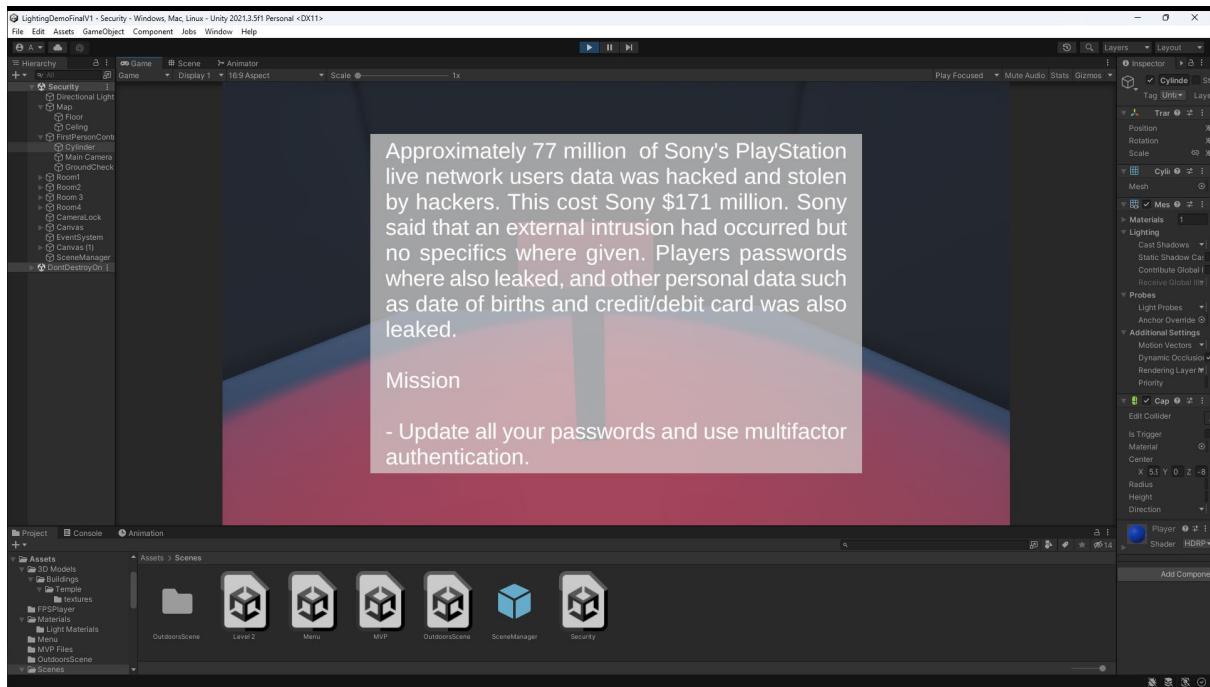


Figure 57 – Dossier

User Interface

The main menu UI has been changed from the wireframe, as there is a total of 3 Levels plus the MVP. This also changed the way the UI was going to look, instead of having a top to bottom list, now it is a left to right list, with images of the levels.

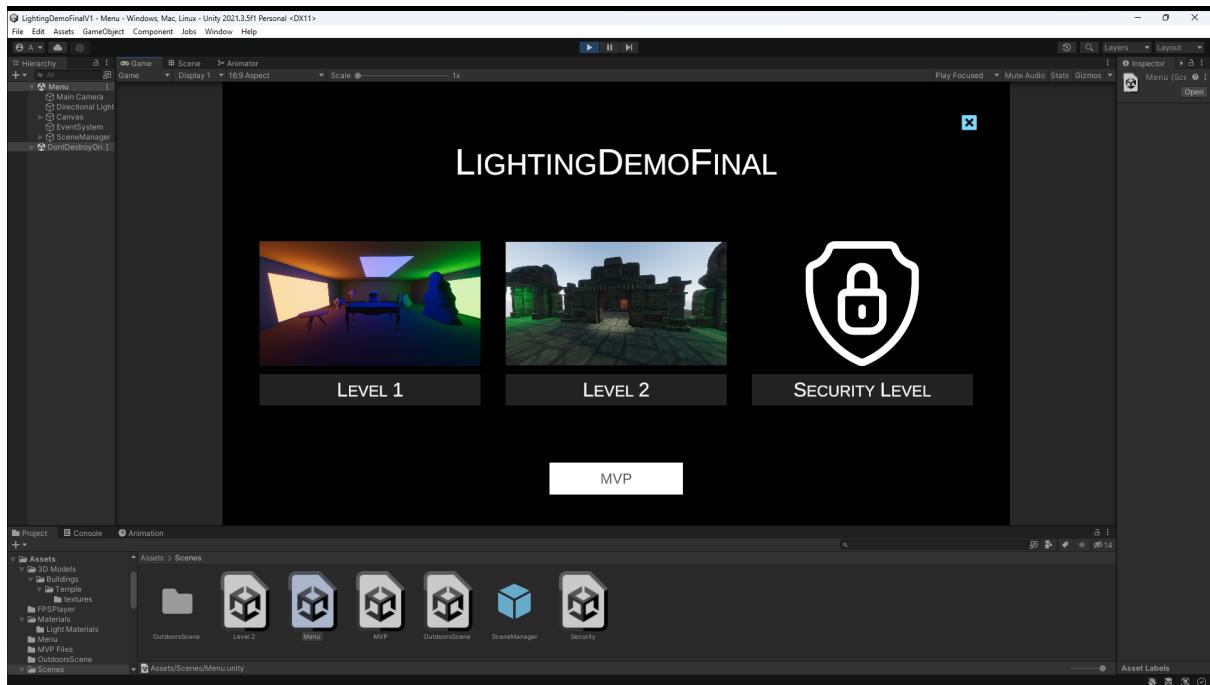


Figure 58 - Main Menu



Figure 59 - Animation Assets

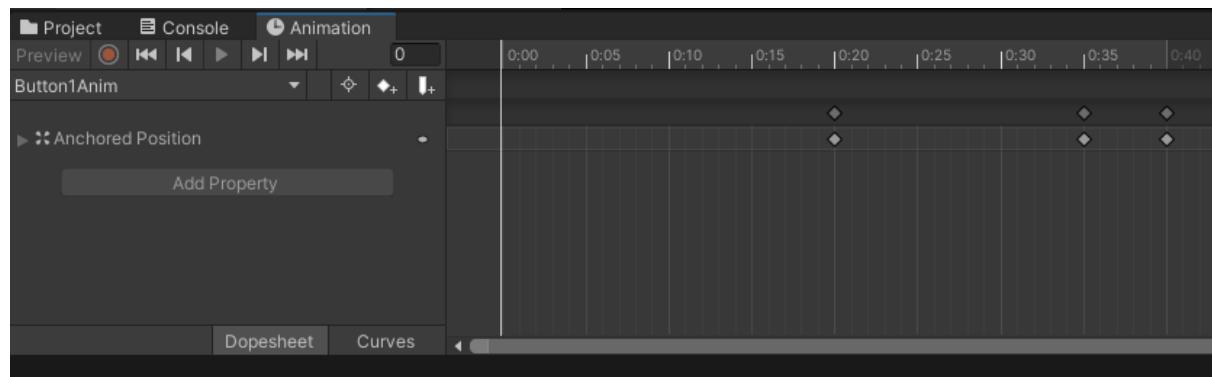


Figure 60 - Keyframes for Animation

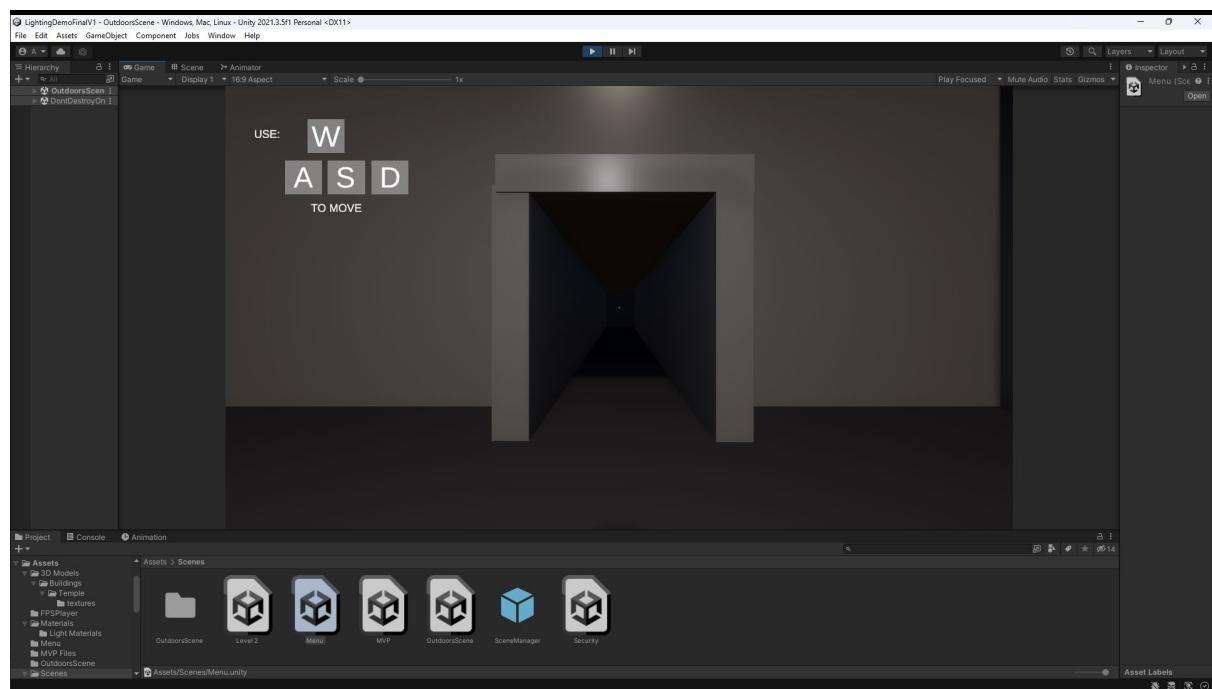


Figure 61 – Movement UI

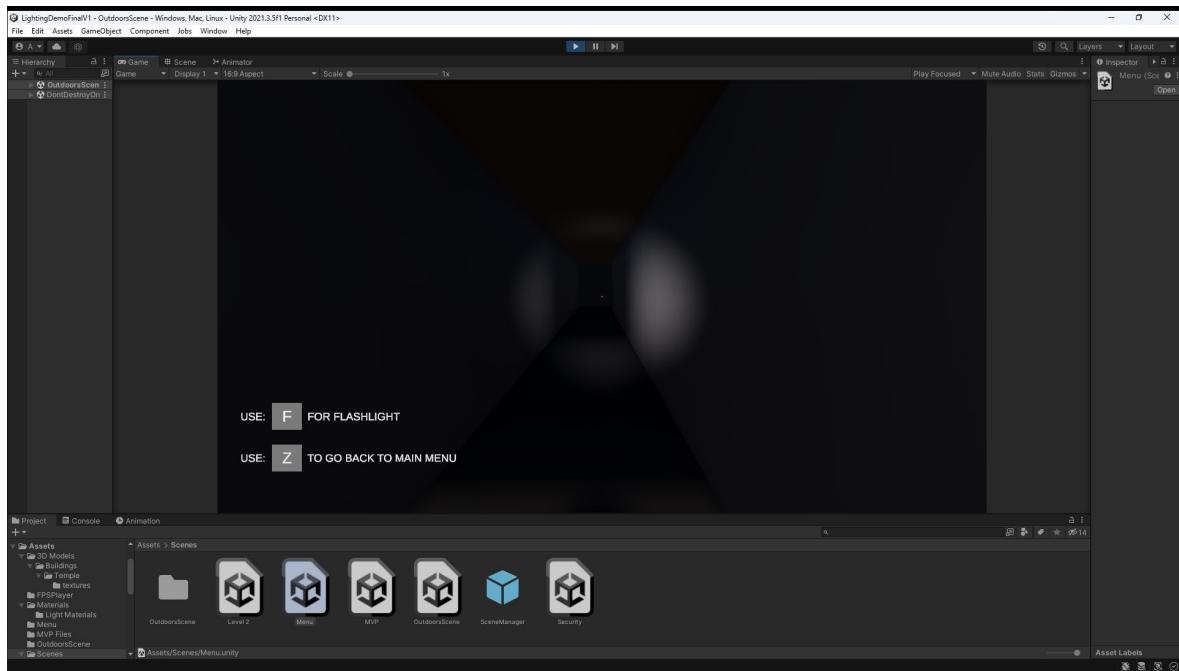


Figure 62 - Other Keys UI

The UI also has many animations that play during gameplay. One animation that the player sees, where the level buttons come up from the bottom of the screen with a bounce.

The other animations in the game are the UI elements like 'WASD' 'F' and 'Z' swipe in from the bottom or top of the screen and then they disappear after a couple of seconds in the game.

These animations were created with Unity's built in animating tool that uses keyframes to make an element move from one location to another (Figure 60). This made it easier to create small effects like the bounce effect on the main menu.

```
public class ZMenu : MonoBehaviour
{
    // Update is called once per frame
    void Update()
    {
        //This if checks to see if the z key is pressed
        if(Input.GetKeyDown(KeyCode.Z)) {
            SceneManager.LoadSceneAsync("Menu"); //Loads the main menu scene
        }
    }
}
```

Figure 63 - LoadScene.cs

```
public class Quit : MonoBehaviour
{
    0 references
    public void Exit()
    {
        Application.Quit(); //Quits the game

        // Just used to see it in editor. Can be removed.
        Debug.Log("Quit!");
    }
}
```

Figure 64 - Quit Button

SECURITY RESEARCH ON IL2CPP

IL2CPP stands for Intermediate Language to C++. IL2CPP improves the performance of Unity games by compiling C# code to native C++ code, which is then compiled into native binary which runs faster than C# code as C++ is a lower-level language, which means it has more access to the hardware. This also allows developers to target machines that don't support C# such as mobiles and consoles (Unity, 2023).

There are many tools on the internet that allow the user to hack unity games. An example of a tool is 'Melon Loader', which allows you to run your own C# code in a unity IL2CPP game. It allows you to enumerate objects into the game and change their properties and call their methods (Hacking, 2023). To see the file structure of a Unity game you can use 'UnityExplorer' which is a mod for Melon Loader that allows the user to see the structure of the Unity game files. It also allows the user to add hooks to the game's method so. The hooks allow the user to modify the method either before running or at runtime.



Figure 65 - Melon Loader

This project is using the mono backend which reduces the risk of getting hacked in this way.

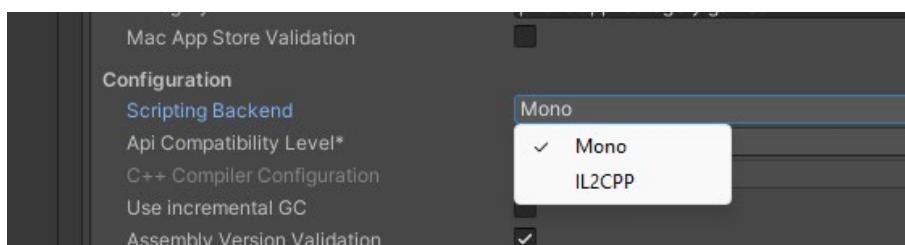


Figure 66 - Mono and IL2CPP s

ISSUES DURING DEVELOPMENT

Technical Issues

The first technical issue was an error in opening a new project once it had been closed once. This crash happened every time the unity project was opened. Unity support was contacted on January 10th. I gave them all the information on what happened and how it happened. I explained to them that when the project opens, it crashes immediately. Even though it is a new HDRP project.

They suggested that the unity editor version should be updated to 2023.1.0a25 and above, but this is an alpha build which cannot be used as it will have bugs and issues because it is an alpha build (Appendix l:).

The issue was fixed with a CPU upgrade (Appendix m:), as the issue was a compiling timeout error. The new CPU is an AMD Ryzen 7 3800x 8 core and 16 thread CPU. After the upgrade, the problem with crashing on startup went away. This issue had lost about 2 weeks of development time as troubleshooting the issue and ordering a new CPU and motherboard took time. This was a big issue as the demonstration of MVPs was due in for the start of February.

The second technical issue that I encountered was near the end of development when I went to test the lighting demo on university computers. The issue was a missed match of the Unity version on University computers, I was making the game on version 2022.2.1f1, and the university computers were running version 2021.3.5f1. The project wouldn't open and there were lots of errors (Appendix o:). This meant that I had to fully rebuild the entire project, which I have done now.

Also, when running the built executable of the game, there were permission problems as my user account didn't have the privilege to load the exe file. I got around this by uploading the file to my OneDrive, then downloading it to the university computers as a zip file. Then I extracted the file. Then the game would load.

Personal Issues

The first issue was in the middle of November, my parents started construction work on the house. The roof needed to be fixed and all the chimneys needed to be removed (Appendix n:). This had a huge impact on the development of the project as my room is the loft, and the roofers were breaking the roof and fixing it all. This made it impossible for me to work on the game project part as I had no access to my pc. I couldn't move the PC anywhere else as the other rooms' chimneys were getting removed. My room was finished at the start of January, which allowed me to bring my PC back up to my room and start the MVP.

The second issue was in the middle of march (16 March), I got ill with the covid-19 virus, due to this I was unable to do any work on the project as I was bed ridden for over a week. This wasted quite a bit of time as it took me a few more days after recovering to get back to normal speed with developing the project.

SUMMARY

This section covered the entire development process and the problems that came up during the development of the lighting demo. It covered the tools that were used in development and the C# programming language. Most of the requirements that were mentioned in the previous section have been met. It also covers the issues during development.

Chapter 7 - Testing and Analysis

This chapter is going to discuss the testing that was done throughout the development of the lighting demo project. It will cover the test plan, what testing methods were used during the testing phase of the project and what the results are.

WHAT IS TESTING IN GAME DEVELOPMENT

Testing in game development is the process of verifying and validating the functionality, performance, and the quality of the game. Testing should be done before release so that any bugs and issues can be found by the game developers. (Scenario, 2020) (Test Bytes, 2022).

Functional Testing

This checks to see if the basic functionality of the game such as the player movement, game mechanics, player interactions, and objectives are working correctly.

Usability Testing

Usability testing involves testing the game's UI, controls, and gameplay mechanics to ensure that they are working correctly and are intuitive and easy to use.

Compatibility Testing

This makes sure that the game can run and play well on different platforms and devices such as Android or IOS devices, different computers, and game consoles.

Performance Testing

The game is tested to make sure that it can handle devices of different levels and for an online game, it's there to check to see if the game can handle lots of players at once.

Play Testing

Play testing is a constant process of running the game live and making sure the flow of the game has not changed from the initial design. This allows the developers to test lots of different elements of the game quickly and efficiently.

WHY IS TESTING IMPORTANT?

Video games are becoming more interactive, with complex gaming mediums and multiple users consuming games across multiple platforms, making them more vulnerable. As a result, testing is critical to avoid player criticism and significant sales losses because of negative experiences.

Quality assessment allows for game developers and game testers to find bugs and glitches in the game and identify a solution to fix.

Testing a game can also show the pitfalls of the game and can show the developers what needs to be upgraded in the game to increase the quality of the overall game. This testing data can also be used for future games.

If a game isn't tested well before a release and it is released with lots of bugs, this can have a huge effect on the game's rank. This will also affect the revenue of the game, as potential customers will look at the bad reviews of the game, and not buy the game (Bultman, headspin).

TEST PLAN

The test plan consisted of continuous testing by continually play-testing the game and continually building the game (Figure 67) to make sure that the game would build with no errors and to make sure that the performance was good enough for the project. The usability was also tested by giving the built game to the tester (Friends and family) and getting feedback from them based on each test, for example, sensitivity of the mouse when looking around.

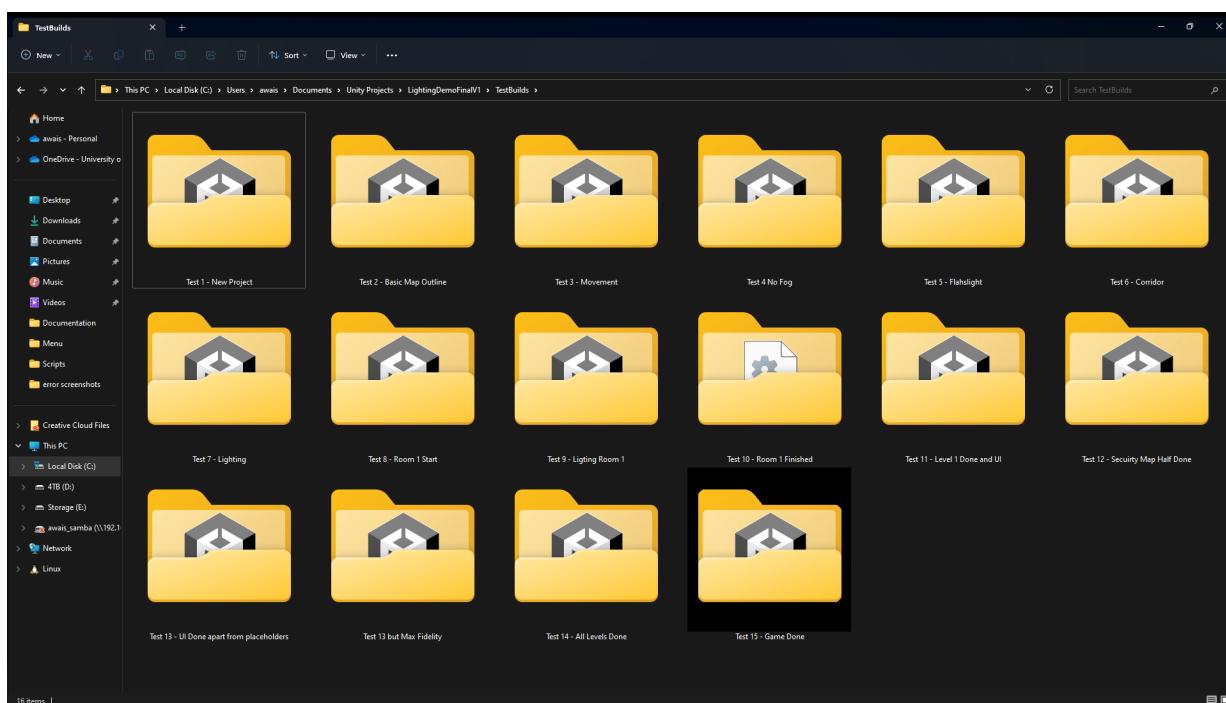


Figure 67 - Build Tests

FUNCTIONAL TESTING

Every feature added was tested straight away to make sure the feature was functioning correctly.

The first feature was the movement mechanics. When testing the mouse's look, the first test that had to be passed was the in-game camera looking around in the game world and having the right sensitivity. The result of this test was that this part worked. The next part to test was the movement. To test this, the player needs to move forward with the 'W' key, move back with the 'S' key, move left with the 'A' key, and move right with the 'D' key, this test passed successfully. The next part to test was the jump mechanics with the ground check, for this test pass the in-game character would need to be able to jump with the space bar, this test was also successful. In the game, there is a check to see if the player is on the ground or not. If the check is true, then the

player can walk forward; if the check is false, the player will fall to the ground. This test was also successfully passed.

The next feature added was the flashlight. The main function of the flashlight is to turn on and off a spotlight when pressing the 'F' key. When the feature was first added, the flashlight would turn on but not turn off. This was then resolved by adding a delay to the keypress, there is a small delay of 0.05 seconds when the key is pressed. This allowed the flashlight to be turned off afterwards.

The next feature that was added was the animation for the menus. This was a tedious process as you must be precise with animation frame as there is a bounce that lasts 0.5 seconds. To get the placement of the bounce correct, it was a lot of trial and error. Eventually the animation was successful.

USABILITY TESTING AND PLAY TESTING

The game was constantly being play-tested during the entire development process. This was to make sure that everything looked good, and the game ran smoothly. And if any errors came up, then they would be quickly dealt with. It is like running an application in visual studio to see if the code added broke the program or not.

The movement controls are based on other first person shooter games like Counter Strike: Global Offensive, or Battlefield 2042. The player moves around with 'WASD' keys, and the player looks around the world with the mouse.

The movement controls were tested by having a family member and friends move around the game world using the keyboard and mouse. The feedback that was given said that controls felt fine and that they were able to go around the game map without any issues. The only negative point that was mentioned was the players' movement felt too quick and that it needed to be slowed down as they couldn't time their jumps correctly. The test was done inside the unity editor so that if any issues came up, they could be fixed quickly.

COMPATIBILITY TESTING AND PERFORMANCE TESTING

The project was tested on 2 windows 11 laptops to see if the build of the game would run on them. The game ran on one of the laptops which had an intel CPU and graphics but not the other laptop that had an AMD CPU and graphics (Figure 68). The project would load in the unity game editor and run at a lower resolution. The issue for this error is that the video memory on the laptop is low and it requires more memory. The laptop only has 1gb of dedicated video memory.

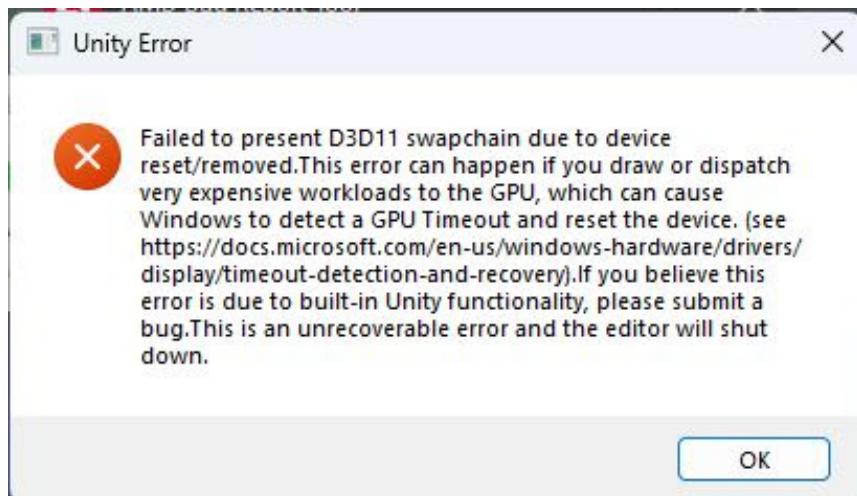


Figure 68 - AMD Laptop Error

The performance on the Intel HP laptop was quite poor when the game build ran, but this was expected as it doesn't have a dedicated graphics card but an integrated one. The game would run fine in the unity editor as the resolution of the game can be reduced.

The main issue that came up near the end of development was that the game wouldn't run on university computers. This wasn't because the versions of the Unity editor where different and the game wouldn't load in the unity editor. The Universities version of Unity was 2021.3.5f1 and the Unity version at home was 2022.2.1f1. There was no easy fix for this issue, the project was then fully ported to the university's version of the Unity editor. This is essentially making the entire game again.

The other test that failed was running the built project on the University computers. This test also failed as the game wouldn't launch due to not having the right permissions. This was fixed by uploading the build to OneDrive and then downloading the application as a zip file and then extracting the build and then running it, then the build would run.

The game again tested on university computers, and this time the game worked in the editor and a build of the game also worked (Figure 69).

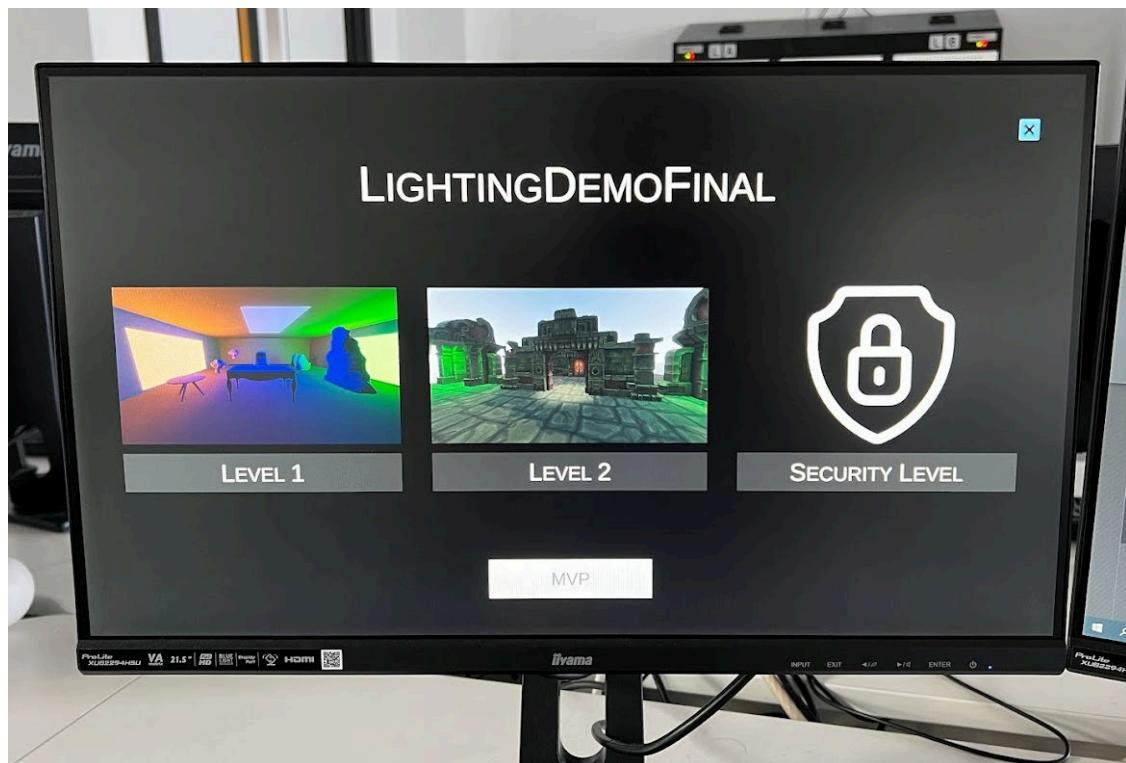


Figure 69 - Game running on University Computers

QUESTIONNAIR AND RESULTS

Below is the questionnaire, and here is a link to the questionnaire -

<https://forms.gle/xbkVw7hrJTPWJbEF9>

Section 1 of 4

Lighting and Rendering in the Unity Game Engine

Consent Form and Questionnaire

Consent Form

The main goal of this questionnaire is to understand if the lighting game demo has achieved the goal of the project which is to show how lighting works in the Unity Game engine. The information gathered from this questionnaire will be used to make improvements to the game demo.

You have the option to take part or not take part in the questionnaire, also if you do accept to take part in the questionnaire, you can decide to leave it whenever you want without any reason

By Signing this form I confirm that:

I agree to participate in this questionnaire.

I have understood the aim of the project, and I have been able to ask questions about the project, and the questions have been answered to my satisfaction.

I am fully aware that the data collected will be stored to the standard of the Data Collection Act of 1998.

Do you wish to participate? *

Yes

No

After section 1 Continue to next section

Section 2 of 4

Personal Details

Please fill in the details.

Please could you enter your full name? *

Short answer text

Please could you enter you email address? *

Short answer text

Today's Date *

Month, day, year 

Figure 70 - Questionnaire Part 1

Section 3 of 4

Questionnaire ✖ ⋮

Description (optional)

1a) When you entered the corridor, did the light effect your decision? *

Yes
 No

1b) If answer to 1a is 'Yes', please could you explain how?
 Long answer text

2) How was the quality of the lighting in the game demo? *

1	2	3	4	5	
Very Bad	<input type="radio"/> Very Good				

3) How was the quality of the graphics in the game demo? *

1	2	3	4	5	
Very Bad	<input type="radio"/> Very Good				

4) How did the movement system feel in the game demo? *

1	2	3	4	5	
Very Bad	<input type="radio"/> Very Good				

5a) Overall, did the demo show the different lighting techniques of the Unity Game Engine? *

1	2	3	4	5	
Does Not Show	<input type="radio"/> Fully Shows				

5b) Please could you explain why you gave the answer for 5a?
 Long answer text

Figure 71 - Questionnaire Part 2

Section 4 of 4

Participation Declined

You have decided not to participate, you can click the submit button or close your web browser.

⋮

Figure 72 - Questionnaire Part 3

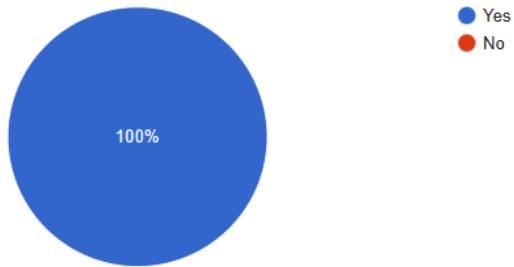
Below are the results of the questionnaire.

Questionnaire

1a) When you entered the corridor, did the light effect your decision?

 Copy

8 responses



1b) If answer to 1a is 'Yes', please could you explain how?

8 responses

I chose to go towards the direction of the light

I went towards light

I moved in the direction of the lights

Went in the direction where there was light

The light made me choose the pathway as i thought it maybe the best option to lead me to somewhere.

It allowed me to see where to go

The light enabled me to decide which way to go preventing me from getting stuck it randomly walking into walls. The light made me able to see what direction to go.

I couldn't see anything on the other side

2) How was the quality of the lighting in the game demo?

 Copy

8 responses

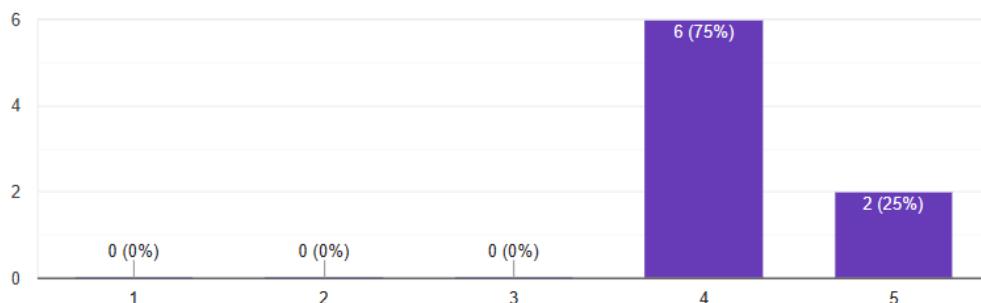
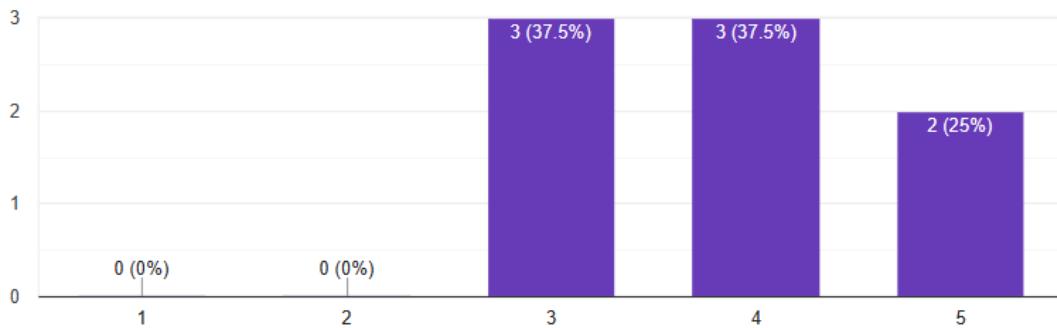


Figure 73 - Results part 1

3) How was the quality of the graphics in the game demo?

 Copy

8 responses



4) How did the movement system feel in the game demo?

 Copy

8 responses

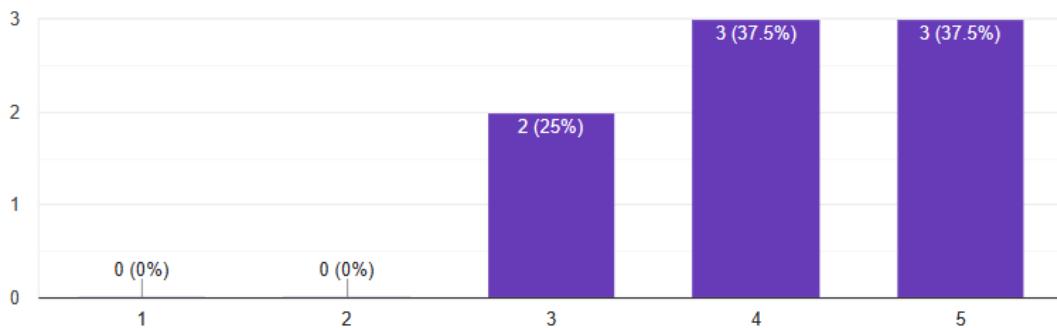


Figure 74 - Results part 2

5a) Overall, did the demo show the different lighting techniques of the Unity Game Engine?

8 responses

Rating	Count	Percentage
1	0	0%
2	0	0%
3	0	0%
4	6	75%
5	2	25%

5b) Please could you explain why you gave the answer for 5a?

8 responses

Yes overall it showed all the different types of lighting in the demo and it ran smoothly with no issues.

Yes as it shows how good the lighting system is in unity and how effected my decisions when picking where to go.

It showed glowing lights, lights that could be toggled and the effect of lighting on the environment

It showed the effects of light coming from multiple sources in dark indoor environments and outdoor environments

There was shadows and it looked very similar to what you would see in real life.

Yes I did, I could see how the lighting was different for the outside scenes

I was able to move the person around successfully with no glitches and it did not freeze. Loading up time was good. I was also able to use all accessories to the demo but was very limited. Demo also shows the different lighting techniques.

Figure 75 - Results part 3

SUMMARY

The testing of the project with lots of different methods was successful, the game worked, and all the builds worked and ran on the development computer and the University computers as well. The people that answered the questionnaire also showed that the project is successful and meets the goals of the project.

Chapter 8 - Critical Evaluation

This chapter is going to contain a personal evaluation of the entire project. From meeting the objectives to the strengths and weaknesses of the final product. It will also detail the lessons that have been learnt during the project.

OBJECTIVES MET AND DEVIATIONS

No	Objective	Type	Met	Reason
1	Learn how to use the Unity 3D game engine by watching unity tutorials on YouTube then make a game map.	Core	Yes	
2	Make 3D models using Blender by watching Blender Guru on YouTube and make a character model.	Core	Yes	
3	Learn how to code in C# by watching a YouTube tutorial and make a first-person controller script.	Core	Yes	
4	Add sounds in the game by looking.	Core	No	Due to construction work.
5	Learn how lighting works in unity by using the Unity Engine's.	Core	Yes	
6	Make a minimal viable product that has a simple scene, user interface and game mechanics.	Core	Yes	
7	Add sounds to the game, also add music to the games.	Core	No	Due to construction work.
8	Add good lighting to the game.	Core	Yes	
9	Fix all the bugs in the final part of the game.	Core	Yes	
10	User testing with people so they can give feedback to me about the game.	Core	Yes	
11	Research Ray Tracing to see what it is and how it works.	Optional	No	Due to illness and not having the graphics power.
12	Research DRM and security measures and try to add it to the game.	Optional	Yes	

Table 1 - Objectives

Objectives 4 and 7 were dropped due to the construction work mentioned in the development and implementation section. This feature wasn't critical to the project's completion as the main aim for the project is based on lighting.

Optional objective 1 was also dropped due to illness that was mentioned in the development and implementation section. Also, another reason for not completing this objective is that I didn't have the correct hardware to meet this goal.

The key objectives that need to be met are based on the lighting and graphics and movement. These objectives have been fully developed and fully tested.

LESSONS LEARNT

I have had the opportunity to use my skills from my time at the University and from my industrial placement, on this project. It has also taught me many lessons during this final year project.

One lesson that I have learnt is that I need to look at the small details of everything to make sure I don't miss anything. The mistake I made was a small one, I made my first version of the project with a different version of Unity than the one that was installed on the University computers. This mistake made it so that the project wouldn't load on the university computer. I had to remake the entire project with the University version of Unity. This lesson taught me how to quickly solve problems to meet deadlines.

Other lessons that I learnt during the year were how to manage my time and know how to understand a project and figure out what the important parts of the project are and what are not that critical to the project. This was very important for this project as I had a few issues that reduced the time I had to develop the project and I needed to cut parts of the project to meet deadlines.

This project has taught me how to quickly teach myself new topics that I have never learnt before. It has also taught me how to get good and legitimate resources to read or watch to increase my understanding of a topic.

REFLECTIONS – DELIVERABLES

Overall, the first deliverables were excellent, and nothing needs to be changed as it gave me directions to follow for the project. The only thing that could be better is the Gantt chart as it didn't account for issues that could come up.

The second delivery was also excellent and gave me directions to follow. The methodology chosen was agile, and it was the correct choice looking back. The only part I would change would be the render pipeline that I chose. The HDRP pipeline is good for high quality graphics, but it made it harder to test on lower end hardware. If I did the project again, I would choose to use the URP pipeline.

Overall, I am happy with the deliverables as they gave me directions to go towards and gave me targets to hit.

SUMMARY

The project was worth the time and effort. The project has taught me many things about time management and how to deal with issues that are out of my control. It has increased my understanding of the importance of lighting in games and has shown me how important security is in the gaming industry.

Chapter 9 - Conclusion

The project and dissertation set out to find out how lighting works in the unity game engine and how it can affect the player when playing the game and what security issues there are in the gaming industry. Using the agile methodology and other tools, this project has successfully shown how Unity handles lighting in the engine by using a gaming demo to show the different lighting effects. This project has also shown the ways developers use light to affect the decision making of players by making a map that uses lights to guide the player. This project has also shown the threats to players and companies in the gaming industry by having a level that informs the player of the threats and what the player can do to protect themselves from these threats.

The future of lighting is using raytraced graphics to give a real life look on lighting. Unfortunately, my project doesn't use any ray-tracing techniques to render the lighting as I didn't have the hardware to develop this. Nvidia has many technical demos that use raytracing to get life like lighting and graphics, an example of this is Marbles at night tech demo (Nvidia Geforce, 2020). If I was to do this project again, I would focus more on ray-traced lighting as well as the normal lighting techniques in Unity. Also, I would stick to one level that shows all the techniques working together instead of separate levels that show the individual techniques.

Overall, I believe this project was a success as it hit the main objectives and the feedback from the users that tested the project was also excellent and there were a few issues to fix during the development of the project.

References

- 1-up. (2021, March 08). *Turning Movement into Gameplay*. Retrieved from Medium: <https://medium.com/@1-UP/turning-movement-into-gameplay-video-game-systems-428791dac211#:~:text=Movement%20systems%20are%20the%20means,environmental%20obstacles%20in%20their%20way>.
- Alexander, A. (2018, May 11). *Working on Lighting for Video Games*. Retrieved from 8olv: <https://80.lv/articles/working-on-lighting-for-video-games/>
- BARI, M. A. (2023, February 7). *GTA Online PC Exploit Allows Hackers To Access IP Address, Rockstar Looking Into It*. Retrieved from Twisted Voxel: <https://twistedvoxel.com/gta-v-gta-online-pc-update-online-security-fix-exploit/>
- BitGem. (2017, 27 August). *Mayan Temple*. Retrieved from DevAssets: <https://devassets.com/assets/mayan-temple/>
- Blender Foundation. (n.d.). <https://www.blender.org/about/#:~:text=Blender%20is%20the%20free%20and,video%20editing%20and%20game%20creation>. Retrieved from Blender: <https://www.blender.org/about/#:~:text=Blender%20is%20the%20free%20and,video%20editing%20and%20game%20creation>.
- Brackeys. (2012, December 22). *Brackeys*. Retrieved from Youtube: <https://www.youtube.com/@Brackeys/videos>
- Brackeys. (2019, October 27). *FIRST PERSON MOVEMENT in Unity - FPS Controller*. Retrieved from Youtube: https://www.youtube.com/watch?v=_QajrabyTJc&t=128s
- Brackeys. (2019, June 23). <https://www.youtube.com/watch?v=MWQv2Bagwgk>. Retrieved from Youtube: <https://www.youtube.com/watch?v=MWQv2Bagwgk>
- Bultman, A. (headspin, June 23). *Aaron Bultman*. Retrieved from headspin: <https://www.headspin.io/blog/game-testing-a-complete-guide-to-its-types-and-processes>
- CAULFIELD, B. (2018, March 19). *What's the Difference Between Ray Tracing and Rasterization?* Retrieved from Nvidia: <https://blogs.nvidia.com/blog/2018/03/19/whats-difference-between-ray-tracing-rasterization/>
- Computer Hope. (2021, December 04). *Computer Hope*. Retrieved from Computer Hope: <https://www.computerhope.com/jargon/b/battle-pass.htm>
- Criddle, C. (2021, February 9). *Cyberpunk 2077 makers CD Projekt hit by ransomware hack*. Retrieved from BBC: <https://www.bbc.co.uk/news/technology-55994787>
- David, A. (2018, July 26). *What are the different methods for testing in game development*. Retrieved from Game Developer: *What are the different methods for testing in game development*

- Donzallaz, P. Y. (2021, January 7). *Explore, learn, and create with the new HDRP Scene template*. Retrieved from Unity Blog: <https://blog.unity.com/engine-platform/explore-learn-create-with-hdrp-scene-template>
- edson-lopes. (2015, September 20). *round table*. Retrieved from cgtrader: <https://www.cgtrader.com/free-3d-models/furniture/table/round-table--8>
- Ellis, N. (2020). *Security and Online Gaming: Understanding Anti-Cheat*. Retrieved from schellman: <https://hub.schellman.com/blog/security-and-online-gaming-understanding-anti-cheat>
- Gemayel, T. (2019, August 12). *How to wireframe*. Retrieved from Figma: <https://www.figma.com/blog/how-to-wireframe/>
- Hacking, G. (2023, February 1). *How to Hack il2cpp Games - MelonLoader Tutorial*. Retrieved from Youtube: <https://www.youtube.com/watch?v=XH4lePNqiHc>
- Helgason, D. (2009, October 28). *A free Unity?* Retrieved from Unity: <https://blog.unity.com/technology/a-free-unity>
- Lucid Software. (2017, July 21). *UML Class Diagram Tutorial*. Retrieved from Youtube: <https://www.youtube.com/watch?v=UI6lqHOVHic>
- Lucid Software. (2018, August 27). <https://www.youtube.com/watch?v=pCK6prSq8aw>. Retrieved from Youtube: <https://www.youtube.com/watch?v=pCK6prSq8aw>
- Lucid Software. (2018, February 17). *UML Use Case Diagram Tutorial*. Retrieved from Youtube: <https://www.youtube.com/watch?v=zid-MV07M-E>
- Mystic. (2019, March 05). *2011 PSN Hack Documentary: How Sony Failed Their Customers*. Retrieved from Youtube: <https://www.youtube.com/watch?v=oBflgU5ComI&t=2s>
- Naiya, D. (2023, March 01). *Call of Duty: Black Ops 3's PC hack letting hackers take control over your device, devs promise a fix*. Retrieved from sportskeeda: <https://www.sportskeeda.com/esports/news-call-duty-black-ops-3-s-pc-hack-letting-hackers-take-control-device-devs-promise-fix>
- Nvidia Geforce. (2020, September 1). *NVIDIA Marbles at Night | RTX Demo*. Retrieved from Youtube: https://www.youtube.com/watch?v=NgcYLIvlp_k
- officetimeline. (2021). *How to make a Gantt chart in Excel*. Retrieved from officetimeline: <https://www.officetimeline.com/gantt-chart/how-to-make/excel>
- Pease, A. (2021, February 8). *A Timely Hack: The Story Of CD Projekt RED's Fall From Grace*. Retrieved from Argon Research: <https://argonresearch.com/cd-projekt-red-hack/#:~:text=On%20February%208th%20CDPR%20announced,a%20dark%20web%20auction%20site.>
- Philips, T. (2016, April 26). *Five years ago today, Sony admitted the great PSN hack*. Retrieved from Euro Gamer: <https://www.eurogamer.net/sony-admitted-the-great-psn-hack-five-years-ago-today>

Pluralsight. (2014, March 15). *Light Up Your World: How Lighting Makes All the Difference for Games*. Retrieved from Pluralsight: <https://www.pluralsight.com/blog/film-games/understanding-the-importance-of-lighting-for-games#:~:text=Light%20has%20a%20drastic%20effect,keep%20the%20player%20on%20edge>.

Pluralsight. (2014, April 22). *What Makes a Good Video Game? 4 Key Elements*. Retrieved from Pluralsight: <https://www.pluralsight.com/blog/film-games/what-makes-a-great-game-the-key-elements-of-successful-games#:~:text=Graphics%20are%20extremely%20important%20for,feel%20of%20the%20game>.

Pluralsight. (2022, November 14). *What is C# prgramming? A beginers guide*. Retrieved from Pluralsight: <https://www.pluralsight.com/blog/software-development/everything-you-need-to-know-about-c#:~:text=When%20compared%20to%20long%20standing,a%20history%20for%20popular%20creations>.

Porter, J. (2021, February 11). *Cyberpunk 2077 studio's hacked data has reportedly been sold*. Retrieved from The Verge: <https://www.theverge.com/2021/2/11/22278121/cd-projekt-red-ransomware-hack-cyberpunk-2077-the-witcher-3-auction-sale>

Raspberry Pi. (n.d.). *FutureLearn*. Retrieved from Layers of an Operating System: [https://www.futurelearn.com/info/courses/computer-systems/o/steps/53514#:~:text=Ring%200%20\(most%20privileged\)%20and,be%20running%20in%20kernel%20mode](https://www.futurelearn.com/info/courses/computer-systems/o/steps/53514#:~:text=Ring%200%20(most%20privileged)%20and,be%20running%20in%20kernel%20mode).

Roach, J. (2020, October 13). *What is DRM in video games and how does it work?* Retrieved from Digital Trends: <https://www.digitaltrends.com/gaming/what-is-drm-in-video-games/>

Run, J. a. (2019, July 01). *Run, Jump and Climb: Designing Fun Movement in Games*. Retrieved from Game Developer: <https://www.gamedeveloper.com/audio/run-jump-and-climb-designing-fun-movement-in-games>

Ryan King Art. (2022, January 31). *Ryan King ASrt*. Retrieved from Youtube: <https://www.youtube.com/watch?v=QHWAbVSkYic&t=86s>

Saylor.org. (2021, June 22). *Software Engineering*. Retrieved from Saylor.org Academy: <https://learn.saylor.org/course/view.php?id=73>

Scenario, T. (2020). *Types of Game Testing*. Retrieved from test scenario: <https://www.testscenario.com/types-of-game-testing/>

Schrute, A. (2019, November 7). *Rendering Pipeline In Unity*. Retrieved from Medium: <https://medium.com/shader-coding-in-unity-from-a-to-z/rendering-pipe-line-fo471aa0904b>

selamcukulatam. (2021, December 06). *Stylized Rock Pack low-poly 3d model*. Retrieved from cgtrader: <https://www.cgtrader.com/items/3097045/download-page>

SomeOrdinaryGamers. (2023, February 20). *Rockstar Games Needs To Fix GTA 5 Immediately...* Retrieved from Youtube: <https://www.youtube.com/watch?v=T4suANqWzc8>

StudioQuadro. (2021, November 19). *free Generic Rock low-poly 3d model*. Retrieved from cgtrader: <https://www.cgtrader.com/free-3d-models/exterior/other/generic-rock-33foo408-6f61-4ca2-acb1-6041cf4862ao>

Tasleem, A. (2022, 10 20). *Logbook*. Retrieved from Google Docs: https://docs.google.com/document/d/1o11IuZP_nEYD1oTxOLe4G3Y3979H4QoLYbTorMfoVLA/edit

Test Bytes. (2022, March 6). *9 Different Types of Game Testing Techniques*. Retrieved from testbytes: <https://www.testbytes.net/blog/types-of-game-testing/>

theflyingtim. (2017, January 25). *Free Office Chair Low poly model low-poly 3d model*. Retrieved from cgtrader: <https://www.cgtrader.com/free-3d-models/furniture/chair/free-office-chair-low-poly-model>

TRAINOR-FOGLEMAN, E. (2021, December 29). *5 timeless principles of good game design for 2022*. Retrieved from Evercast: <https://www.evercast.us/blog/game-design-principles#:~:text=Put%20simply%3B%20good%20game%20design,the%20world%20of%20the%20game.>

Unity. (2016). <https://docs.unity3d.com/540/Documentation/Manual/Lighting.html>. Retrieved from Unity Documentation: <https://docs.unity3d.com/540/Documentation/Manual/Lighting.html>

Unity. (2020, June 05). *Light Mode: Baked*. Retrieved from Unity Documentation: <https://docs.unity3d.com/2019.3/Documentation/Manual/LightMode-Baked.html>

Unity. (2023, March 31). *IL2CPP Overview*. Retrieved from Unity Docs: <https://docs.unity3d.com/Manual/IL2CPP.html>

Unity Technologies. (2021, February 24). *Managed Plug-ins*. Retrieved from Unity docs: <https://docs.unity3d.com/2020.1/Documentation/Manual/UsingDLL.html>

Unity Technologies. (2021). *Unity Visual Scripting*. Retrieved from Unity: <https://unity.com/features/unity-visual-scripting#ready-get-started--2>

Unity Technologies. (2022, November 25). *Glossary*. Retrieved from Unity Docs: <https://docs.unity3d.com/Manual/Glossary.html#globalillumination>

Unity Technologies. (2022, November 18). *High Definition Render Pipeline overview*. Retrieved from Unity Docs: <https://docs.unity3d.com/Packages/com.unity.render-pipelines.high-definition@14.0/manual/index.html>

Unity Technologies. (2022, November 25). *Introduction to Lighting*. Retrieved from Unity Docs: <https://docs.unity3d.com/Manual/LightingInUnity.html>

Unity Technologies. (2022). *Make a 2D game in Unity*. Retrieved from Unity: <https://unity.com/how-to/beginner-2D-game-resources>

Unity Technologies. (2022, November 25). *Reflection Probes*. Retrieved from Unity Docs: <https://docs.unity3d.com/Manual/ReflectionProbes.html>

Unity Technologies. (2022, November 18). *Render Pipelines*. Retrieved from Unity Docs: <https://docs.unity3d.com/Manual/render-pipelines.html>

Unity Technologies. (2022, November 18). *Universal render pipeline overview*. Retrieved from Unity Docs: <https://docs.unity3d.com/Packages/com.unity.render-pipelines.universal@12.1/manual/index.html>

Unreal Engine. (2020, April 29). *UE4 Heightmap Guide: Everything You Need to Know About Landscape Heightmaps for UE4*. Retrieved from World of Level Design: <https://www.worldofleveldesign.com/categories/ue4/landscape-heightmap-guide.php>

Victor. (2014, November 25). *Suzanne Blender Monkey*. Retrieved from sketchfab: <https://sketchfab.com/3d-models/suzanne-blender-monkey-29a3463e8d314c8fbda620800019cfb9>

Appendix a: Project proposal

Introduction

The main aim of this project is to understand how lighting and rendering works in games and how it can help the gameplay and atmosphere for the player. I will also try to learn how the new Unity Render Pipeline works for lighting to see how fast and more optimized games become so that my game can run on lots of older and slower hardware such as older graphics cards like the Nvidia GTX 750ti or even smart phones and use resources on new and faster hardware such as the new Nvidia RTX 4090 in a more efficient way to get higher quality graphics and lighting.

I will hopefully use lighting in a way that enhances the gameplay visually where it can be used in guiding the player or to show items or places of interest and to make the game look more realistic and, I will be able to learn how to optimize my game to run on different platforms.

Who will benefit?

Supervisors

The supervisors will benefit from my project as it will show them the different parts of game development and how each part goes together. It will also show them the security aspect of games and how the security of a game is very important to its success and how a game that is full of cheaters and hackers can kill a player base of the game which will reduce the future potential of the game. It will also show them what the different methods of DRM protection companies use to stop piracy happening of their games.

Gamers

Gamers who will play my game will benefit the most because the lighting will make the game feel high quality and it will also make the graphics look better. They will also be able to use the lighting as a game mechanic to traverse the map. Lighting can also increase the fun factor of the game by allowing the player to use the lighting to their advantage, such as a reflection on a piece of glass to see an enemy or object that is hidden around a corner.

Motivation

I wanted to make a game with good graphics for a long time as I have made games in the past that are on the google play store and on my website ([Awais Tasleem's Portfolio \(awaist7860.github.io\)](#)) These games looked basic and were mostly made for smart phones. The main issue that I had when developing my games was the time needed to make them as I was in high school and college

studying for my GCSE's and A-Levels, I didn't have enough free time to make the games look higher quality and with better graphics. This project will allow me to spend more time on learning more about game development and the different practices that are used in the industry and to improve the skills that I have already learnt over the years whilst developing games to make a higher quality game with better game mechanics.

Objectives

Objective 1

Learn how to use the Unity 3D game engine by watching Brackey's unity tutorials on YouTube then make a game map. This should take me 2 weeks.

Objective 2

Learn how to make 3D models using Blender by watching Blender Guru on YouTube and make a character model. This should take me 2 weeks.

Objective 3

Learn how to code in C# by watching a YouTube tutorial from Barnacules Nerdgasm and make a first-person controller script. This should take me 2 weeks.

Objective 4

Learn how to use Audacity for the music and sounds in the game by looking at video tutorials online and looking at the software's documentation and make sound effects. This may take 2 weeks.

Objective 5

Learn how lighting and rendering works in unity by using the Unity Engine's online documentation and watch tutorials on this topic. This could take me 2 weeks to learn.

Objective 6

With the knowledge from learning the Unity Game engine, I should make a minimal viable product that has a basic scene, a basic user interface and some basic game mechanics, such as camera movement and player interaction with the environment and show this to my supervisor and get feedback on it. My aim for this is to be done by January/February.

Objective 7

Add sounds to the game, such as walking sounds, also add some music to the games and then show it to my supervisor for feedback. This will take a few days to do and my aim for this is the end of January.

Objective 8

From the knowledge learnt from researching the lighting, add good lighting to the game and then show it to my supervisor for feedback. My aim is to get this done and show it to my supervisor by mid-February.

Objective 9

Get a complete game working with all the different parts added together and try to fix as many bugs as possible. I aim to get this done 2 weeks before the demo deadline date.

Objective 10

Show the game to my supervisors and other people so they can give feedback to me about the game.

Optional Objectives

Objective 1

Research Ray Tracing to see what it is and how it works, then I will try adding it to the game.

Objective 2

Research DRM and security measures and try to add it to the game.

Development Requirements

The project will be developed on my pc. The specifications are (May change in the future):

CPU – Intel Core i5 8400 2.8ghz.

GPU – Nvidia GeForce GTX 1060 6gb.

RAM – 16gb DDR4.

OS – Windows 11.

Storage – 500gb SSD and 4tb Hard Drive and 1tb Hard Drive.

Monitor 1 – LG Ultrawide monitor @ 2560 X 1080

Monitor 2 – Lenovo HD Monitor @ 1920 X 1080

The software that I will use to make this project will be, Unity Game Engine for the game mechanics and graphics, Blender for the 3D models that I will need to create, photoshop for making textures and Visual Studio 2022 Community or Visual Studio Code for coding the game in C#. The sound and game music will

be developed using Audacity and (Waveform 12 or LMMS) music software. I will also use Trello.com to organize the project to make sure I know what needs to be worked on. I will use Microsoft's office suite for report writing, making charts, and making presentations.

Methodology

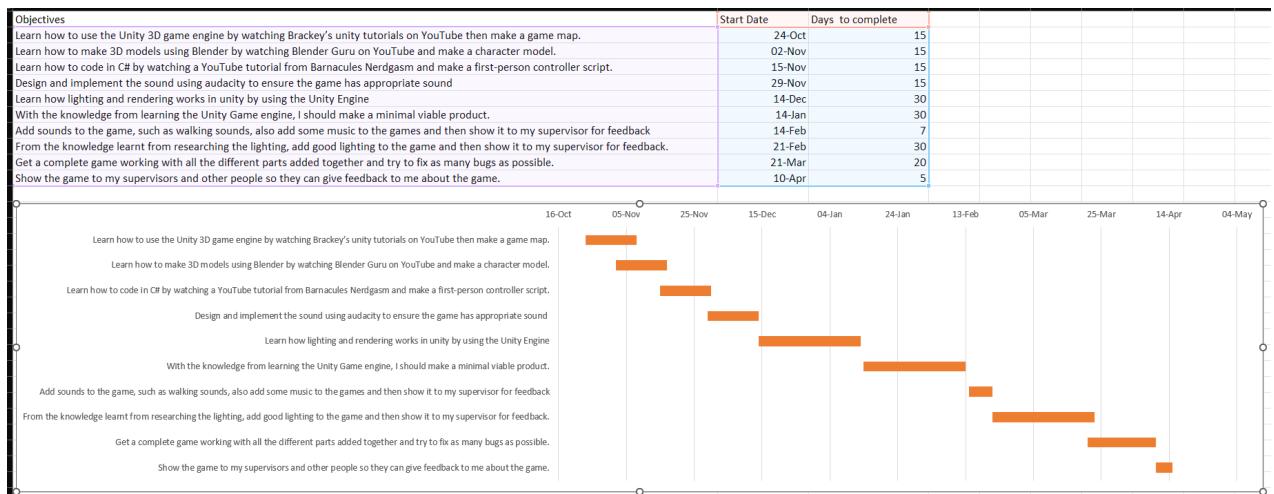
The agile methodology is a way to manage a project by breaking it down into smaller parts and then you work on those parts, once the smaller parts are done, you then combine it all to get the entire project. Many game development studios use these methodologies to make their games. You must continuously talk with your stake holder or supervisor to get feedback from them in the project to make sure you are meeting their requirements at every stage of the methodology. The main parts for this methodology are planning, executing, and then evaluating.

The agile methodology isn't just a single framework, it has many frameworks within it such as scrum, extreme programming, and Kanban. For this project, I will be using the scrum framework to complete the game. Scrum allows the developer to address complex and constantly changing problems, whilst allowing them to deliver the highest quality product that they can make.

As I am developing my game, I will be working on a small part of it at a time until I get to the final product. This way of working is better and less complex than working on a big part or multiple parts at the same time, as this will make it harder for me to test and it will be harder to debug when issues come up.

I will instead follow the scrum methodology and break the problem down into smaller steps. I will plan what needs to be done, this will allow me to stay on track on what needs to be done and not waste time on what doesn't need to be done. Then I will work on the step either by coding or making a 3D model or making some game sounds. Once this is done, then it will be tested and then shown to my supervisor for feedback. Then I will repeat this until the game's made.

Gantt Chart



Appendix b: Logbook

Semester 1

Week 1 (3 Hours)

19/09/2022

Dr Norman Murray gave a lecture about the final year project and how to work on it.

The final year project is worth 40 credits which is 1 third of the final year.

Dr Norman talked about how to make a good start to the project, he suggested that we read the project guide properly, and that we should scope the project properly and that we need to make a realistic project plan. He also said that we should give this project about 2 days a week to work on as it is worth a lot of credit. He also told us to keep our logbook updated.

He talked about the project life cycle which includes:

Choosing the project

Managing the project

Doing the project

Presenting the project

He then talked about how to propose our own project. To do this, the project needs to be challenging to do, must be realistic to do, produce a software/website,

Identify the customer, provide a title, description and a plan and then find a member of staff to act as a supervisor.

Dr Norman also talked about the risk assessment form that we need to put in by the end of week 5. This is a requirement from the university so that we can do the final year project. I need to read, understand, complete, and be signed by me and my supervisor.

Dr Norman also talked about the Ethical approval form that we also need to put in by the end of week 5. This is another requirement from the university so that we can do the final year project. I need to read, understand, complete, and be signed by me and my supervisor.

The final year project module has a few components that need to be done throughout the year, these are:

Project proposal

Introduction, Literature Review and Methodology

Requirements, Design and Implementation, Testing and Evaluation, Discussion, Critical Evaluation, Reflection

Project Presentation and Final Product

Dr Norman also talked about how to avoid plagiarism and what the consequences of plagiarism are.

Independent Research / Study

I was on a placement year last year at brakes manufacturing company called Knorr-Bremse. My job role was IT Placement student. I had learnt a lot about managing projects, understanding the workload, figuring out where to get help, and how to work on my own in a professional manner. I think the skills that I have learnt will be very beneficial to me while working on the Final Year Project.

The change from working a job and going back to studying has been a difficult one for me, as I haven't done much studying outside of work as I was living on my own and had other responsibilities to manage.

The idea for my project was not finalized yet, I knew that I wanted to do a game development project, but I was not sure on how to start it and what I needed to do. I took advantage of the trial period of modules and went to Dr Normans Lecture on game development, he gave me a few ideas on how to make a successful game project. One of the ideas that he gave that stood out to me was the idea about a game based around lighting.

I also went to Dr Ian Drumm's lecture on mobile development to see if making a game for a mobile device would be feasible for me, he suggested that I could use the unity game engine to develop games for mobile.

Week 2 (8 hours)

26/09/2020 – 30/09/2022

Dr Norman Murray gave a lecture about the project proposal.

He told us that the project proposal needs to be no more than 1500 words, and it needs to be appended to this logbook.

The project is important as it reduces the risk of the final product being different than the supervisor expected. It also helps us to keep our focus on the important goals.

Dr Norman gave us a structure of the project proposal:

Introduction

What are you going to research/develop?

Who is it for? Who will benefit?

Why are you doing this?

How are you going to do it?

Objectives

Gantt Chart of objectives to show timeliness and all your deadlines.

Development requirements/ methodology

Logbook

Presented well and written in competent technical English.

Dr Norman also talked about how our objective should be smart objectives:

Specific, states exactly what you need to achieve.

Measurable, includes a quality or quantity measure.

Agreed, between you and your supervisor.

Realistic, can be challenging but must be achievable.

Timebound, with a clear end date or timescale

Dr Norman also talked about which methodologies we need to use, and what the hardware requirements are. He also talked about the end of project presentation and what standard it should be made at.

He also gave a reminder to complete the risk assessment and ethical form.

Independent Research / Study

This week I started doing research on the unity game engine. The research led me to the Unity Render Pipeline, and that there are 2 pipelines, which are Universal Render Pipeline (URP) and High-Definition Render Pipeline (HDRP).

I also started to watch tutorials from Brackeys about making a game with the Unity game engine. He has many useful tutorials that are very beginner friendly.

Week 3 (8 hours)

03/10/2022 – 07/10/2022

Dr Julian Bass gave a lecture about project development methods.

Dr Julian talked about the reason for having the final year project, which is to meet the requirement for the BCS chartered institute of IT. He also talked about the Agile methodology in software development and how it works and why it is needed. The reason we use a methodology in software development is that it maximizes productivity and reduces risk.

Independent Research / Study

Idea for lighting happened here.

The research from the previous week has now allowed me to pick a topic for the game development project. The idea is to make a unity a lighting game. This would be a good way to show off how lighting works in games. I also decided to use the HDRP pipeline for this project as it should give me the best result for the lighting in the game.

This week I also started to plan my project proposal as it was due on week 5 21st October, and I have my other modules to work on as well.

Week 4 (6 hours)

10/10/2022 – 14/10/20200

Dr Chris Hughes gave a lecture about research methods.

Research is the study of people for the purpose of increasing your understanding or adding to knowledge.

Students do the final year project to allow them to develop their own thoughts, ideas, and concepts so that they can question and look at things in new ways. It also allows the student to work on multiple modules at the same time. It also allows students to develop new skills. The student can contribute to the topic.

The processes of research are:

Review the Field.

Build a theory/hypothesis.

Test the theory/hypothesis.

Reflection and integration

Dr Chris talked to us about the literature review that we need to complete for the final year project. The literature review acts like an introduction to your project, it gives a reason to why you are doing the project, it shows the past and current research on your topic, and it also provides you supervisor and other researchers a starting point from where your project started from.

Independent Research / Study

My focus for this week was to get the first draft of the project proposal done and email my supervisors for the project. The first thing that I focused on was the Who will benefit part, this is an important part as it outlines why the project should exist.

After that, I wrote up the development requirements for the project. This was important because the Unity Game engine requires a lot of GPU and CPU power to run.

Then I wrote up a draft of my objectives and optional objectives. After that I wrote up my methodology for the project. For this project I think I am going to work with the agile methodology.

Then I wrote an Introduction to the document and then added this logbook to the proposal by using a link and then I made the entire project look professional.

Week 5 (8 hours)

17/10/2022 – 21/10/2022

Lee Griffiths gave a lecture about Requirements engineering.

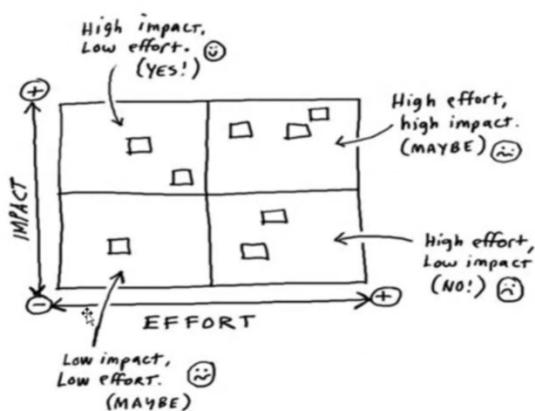
The summary of this lecture is that we should talk to the people that will use the software, understand what exactly needs to be built, do not over promise and under deliver, go back and talk again and keep them in the loop of what is happening and test the software with them. Lee also talked about the role of requirements in software projects that failed. It showed us the importance of understanding the user requirements.

Lee showed us the relative cost of fixing errors in a software application. The highest cost is when the application has been released and the software still has errors.

He also showed us the relative cost of software development stages, and the highest cost was maintaining the software after it was released.

He showed us a box diagram of effort to impact in software development.

Here is a picture of it:



Independent Research / Study

Once I got feedback from my supervisors, I updated my project proposal with the feedback that was given to me.

The main feedback that I was given was that my aims were not too clear. The aims needed to be milestones; learning isn't the only objective.

I have now submitted my proposal for marking.

Week 6 (6 hours)

24/10/2022 – 28/10/2022

Julian Bass gave a lecture about the introduction and literature review about our final year project. This was the first part of the dissertation that had to be done and it will be assessed.

Julian Bass talked about the different approaches to writing, how the different parts should be composed like sentences and paragraphs, how to write a good introduction, literature review and methodology.

Julian Bass explained the importance of writing multiple drafts and having them checked repeatedly. He also emphasized the use of spell checkers and grammar checkers. He also talked about the main goals of the minimal viable product.

Independent Research / Study

This week, I was informed about the risk assessment and that it needed to be done for my final year project.

I've had an issue with the risk assessment, after submitting the risk assessment, there was an issue where my supervisor couldn't see the risk assessment. This issue still hasn't been resolved. This has wasted quite a bit of time for me.

Week 7 (6 hours)

31/10/2022 – 04/11/2022

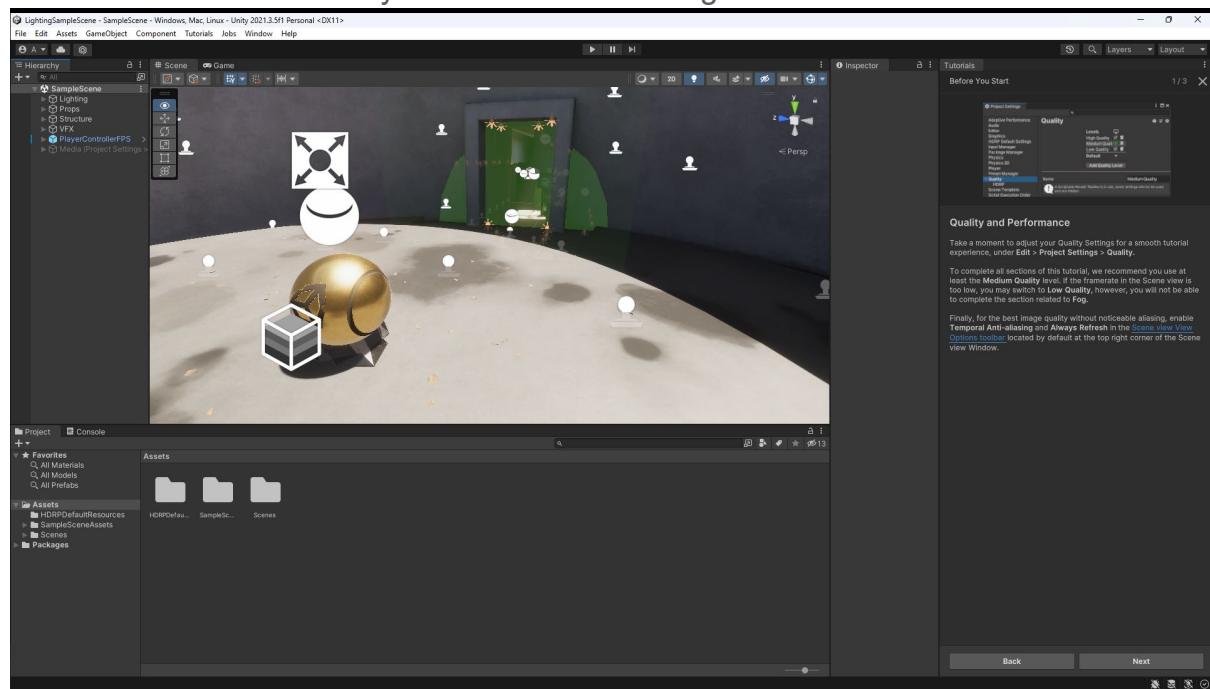
Lee Griffiths gave a lecture about requirements engineering this week.

The FYP lecture by Lee Griffiths focused on gathering requirements engineering for a project and ensuring that requirements meet customers' expectations. Lee emphasized the importance of not being over-ambitious in terms of features and consistently evaluating the software. The lecture highlighted a model that demonstrated the role of poor requirements and communication in software project failures. Lee discussed the relative costs of software development stages and the importance of design in the development process. Lee finished the lecture by giving tips on what to discuss within requirements of the deliverables. If this lecture were given earlier, it would have been helpful, as I have already submitted my project proposal.

Independent Research / Study

This week the risk assessment issue got resolved. My supervisor is going to check the email attachment one instead.

I have now also started a Unity HDRP Lighting tutorial. It is one that is built into the Unity engine editor.



So far, the tutorial seems to be easy to follow, but there is quite a lot to go through.

Week 8 Last Lecture of semester 1 (5 hours)

07/11/2022 – 11/11/2022

Ian Drumm gave a lecture this week about software architecture.

Dr Ian Drum's lecture on software architecture provided useful insights for my lighting project. Ian discussed the development of abstraction techniques in computer science and various architectural styles, along with their advantages and disadvantages. He also discussed various levels of design, such as OOP and MVC, and client-server systems. The lecture provided me with a better understanding of different architectural designs that I can implement into my final year project. Overall, the lecture was helpful and gave me lots of valuable information to consider.

Independent Research / Study

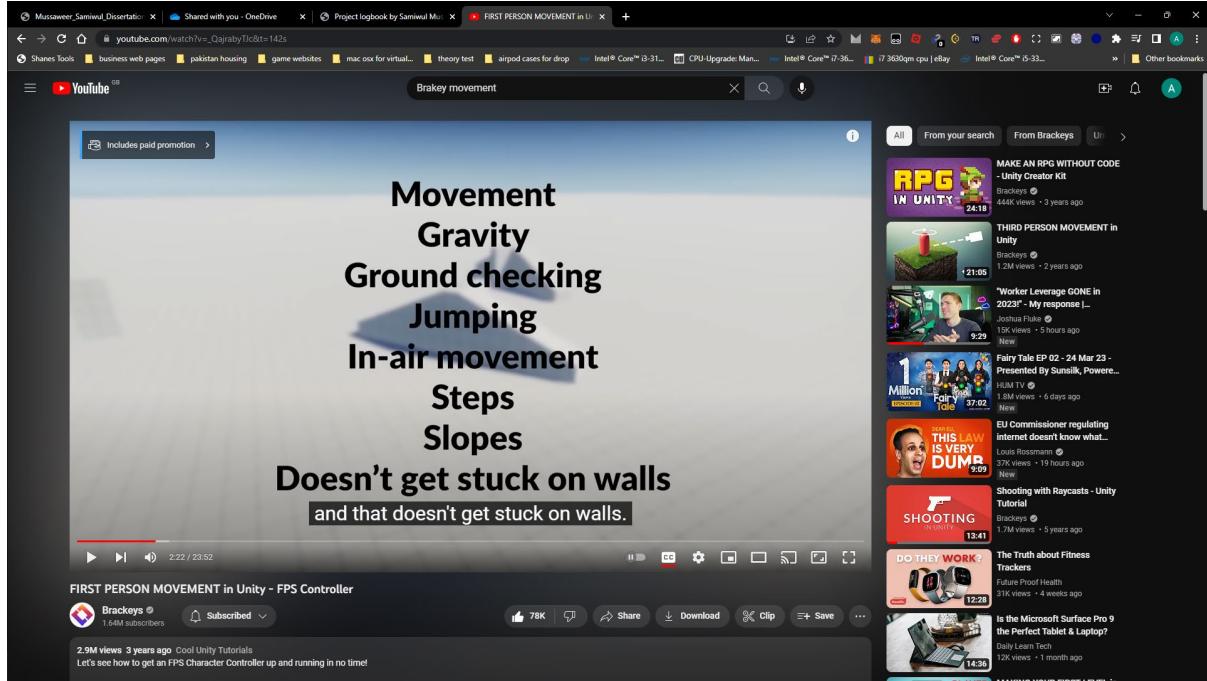
I have now finished the HDRP tutorials, and I am now watching a Brackeys tutorial to gain more knowledge on game design, map design and character movement.

Week 9 -no more lectures (2 hours)

14/11/2022 – 18/11/2022

Independent Research / Study

At the start of this week, I was following a Brackeys tutorial on movement.



Halfway through the week, my parents decided that the roof needed to be done and they needed to remove all the chimneys in the house. For this reason, I was unable to finish the tutorial as my room is the loft and that was the first room affected. I do not have access to my computer anymore, I am currently working on an iPad to do more research.

Week 10 - no more lectures (8 hours)

21/11/2022 – 25/11/2022

Independent Research / Study

Construction work

I was doing research on the process of game development, I have been watching videos and reading articles on this as well, I was taking notes in a OneNote file.

I have also started work on the first draft of the literature review and methodology. So far, I have completed the Lighting section, the Rendering Pipeline in Unity, HDRP vs URP and DRM and security issues game development.

Week 11 - no more lectures (8 hours)

28/11/2022 – 02-12-2022

Independent Research / Study

Construction work

This week I have fully completed the Literature review and methodology. I have talked about the methodology that is going to be used, the programming tools that are needed. And a summary and introduction. I have also emailed the first draft to my supervisors as well.

They have provided feedback on the report, the feedback was that I need to focus a bit more on the security side of the project.

Week 12 - no more lectures (1 hour)

05/12/2022 – 09/12/2022

Independent Research / Study

Construction work

I submitted the assignment on Monday. I have not done any more work on the project as I have other deadlines for other projects that need to be done as well.

Literature review and Methodology Submission 05/12/2022

Week 13 – no more lectures

12/12/2022 – 16/12/2022

Independent Research / Study

Construction work

I have not done any more work on the project as I have other deadlines for other projects that need to be done as well.

4 Weeks Christmas Vacation – 17/12/2022 – 15/01/2023

Issue with Unity contacted Unity Support.

Semester 2

Week 1 (4 hours)

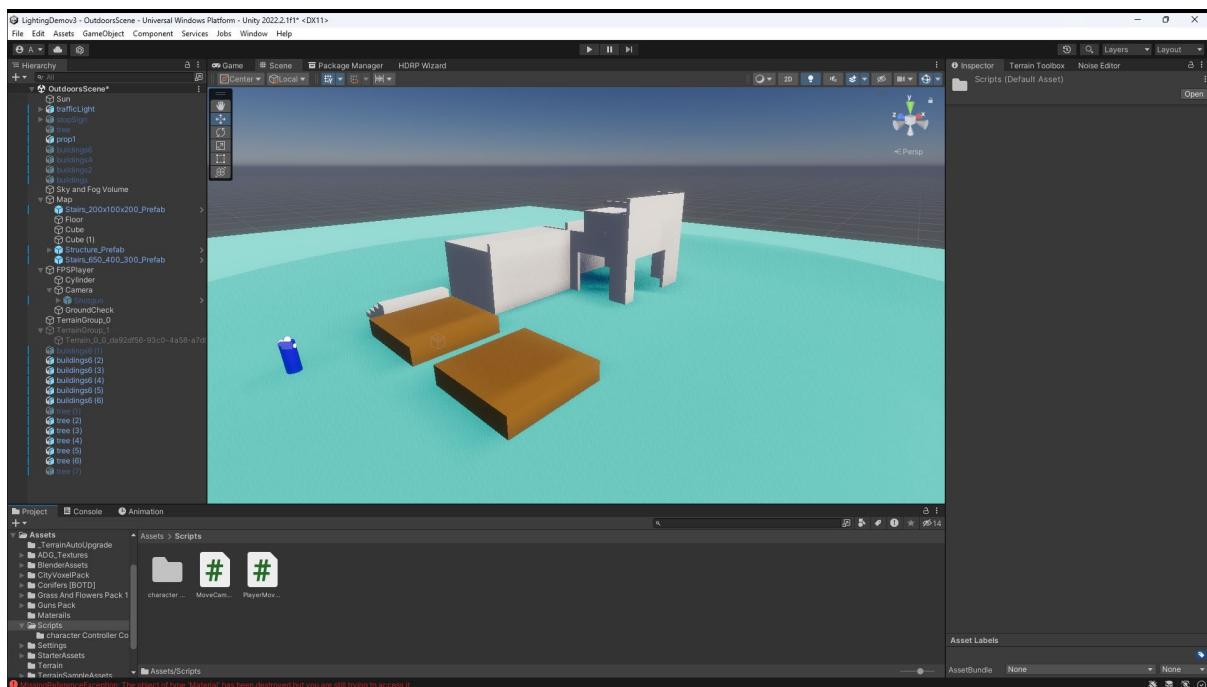
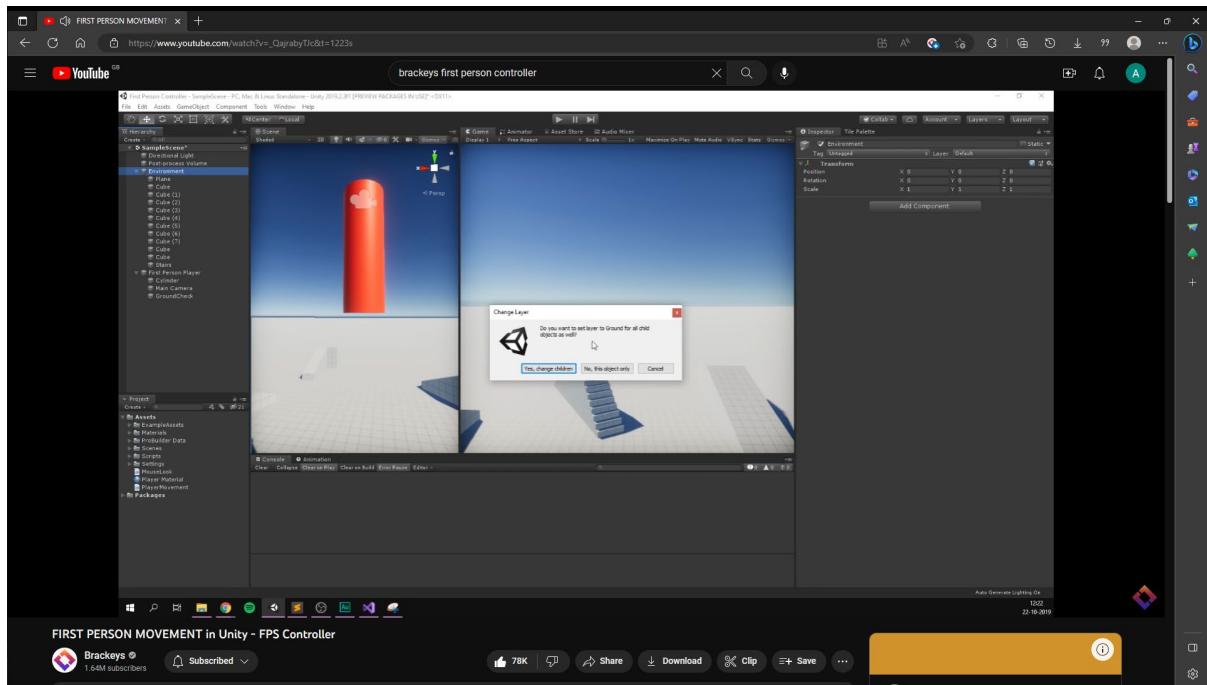
16/01/2023 – 20/01/2023

Independent Research / Study

The construction work finished on the 8th of January 2023. I am now able to use my PC again to develop the project instead of just doing research on a laptop.

Started the MVP demo.

I have gotten a basic map made with some basic unity models. I have also coded a basic character controller so that I can go around the map. I followed Brackeys tutorial to make the controller work.



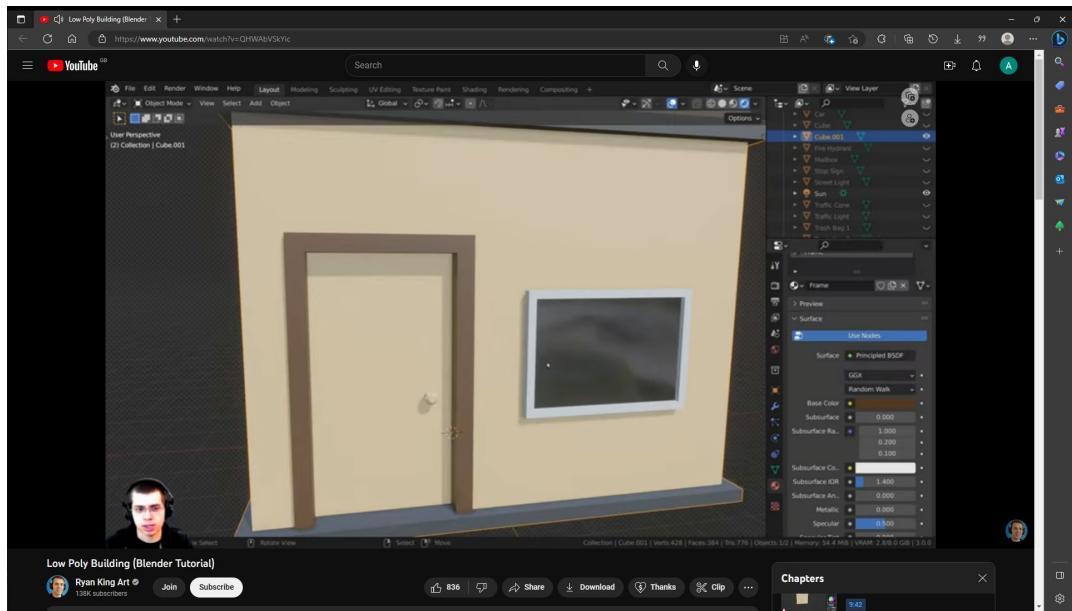
Week 2 (6)

23/01/2023 – 27/01/2023

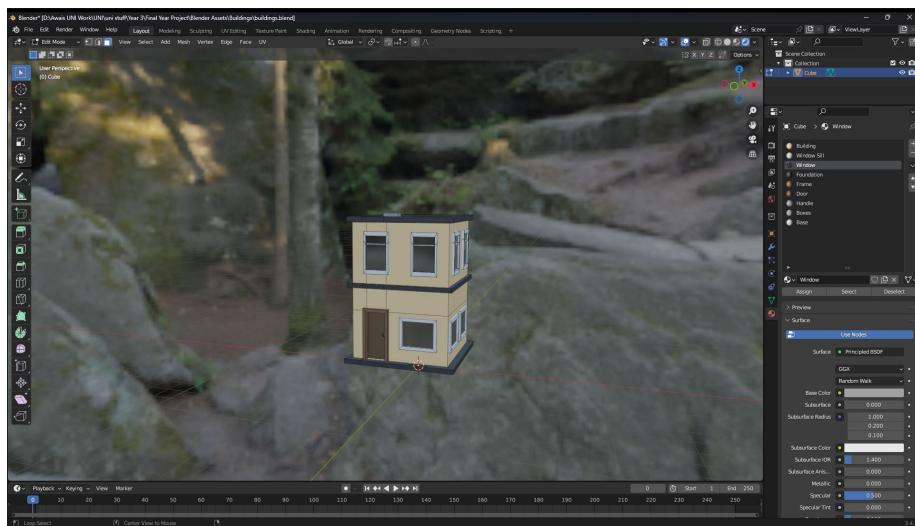
Independent Research / Study

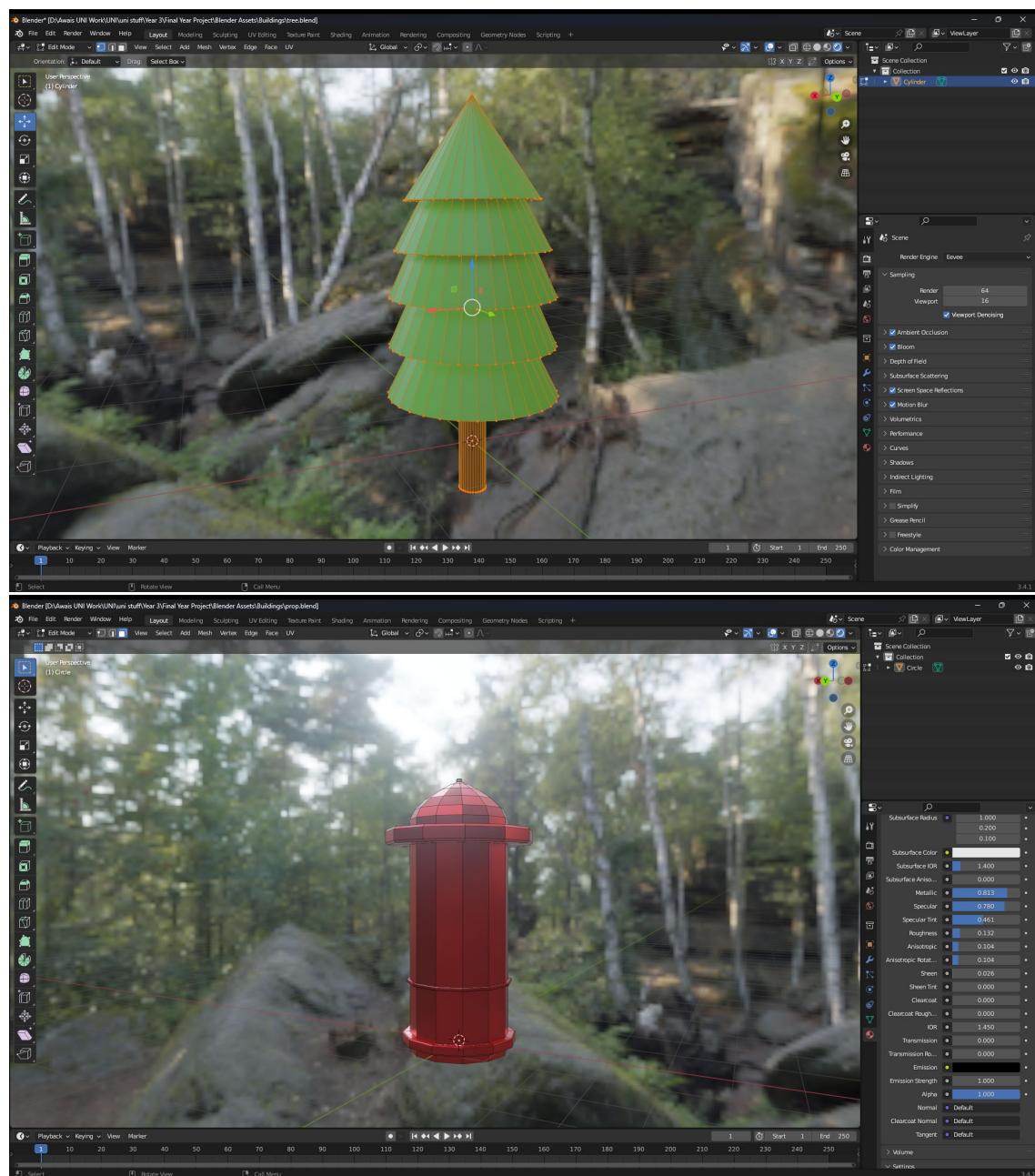
Started making blender models.

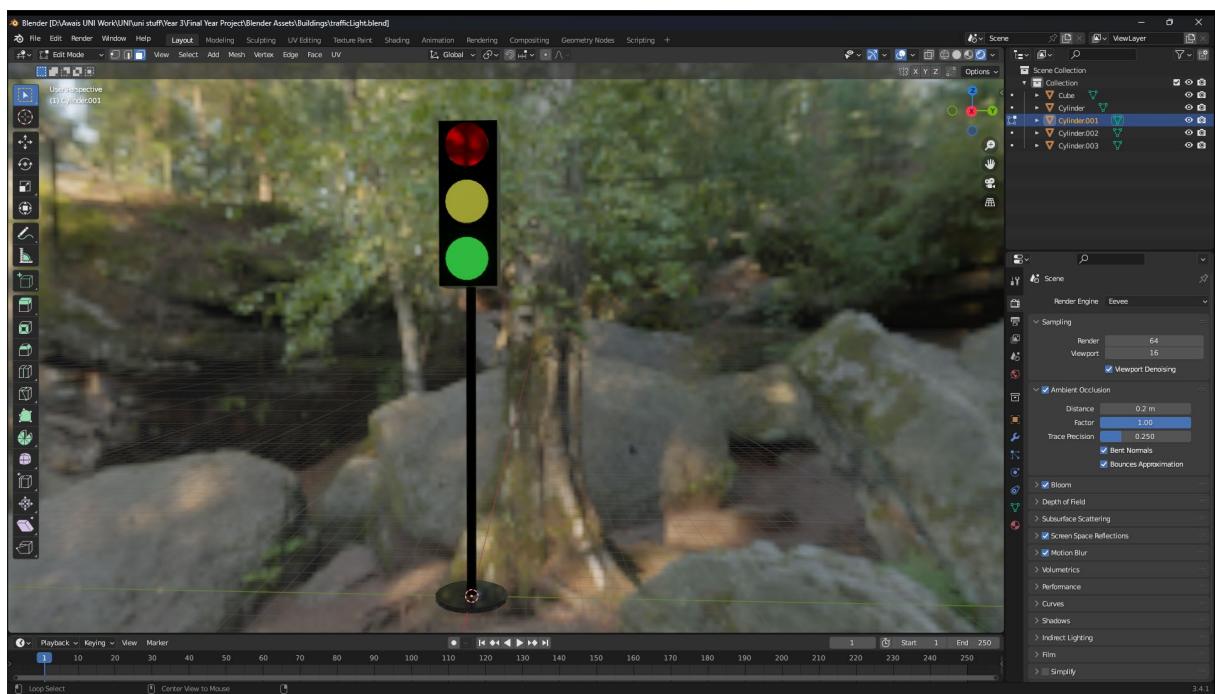
I followed a YouTube video on making a low poly building in blender.



After making the building myself, I then made many other models by looking online for images and then remaking them by following the images.







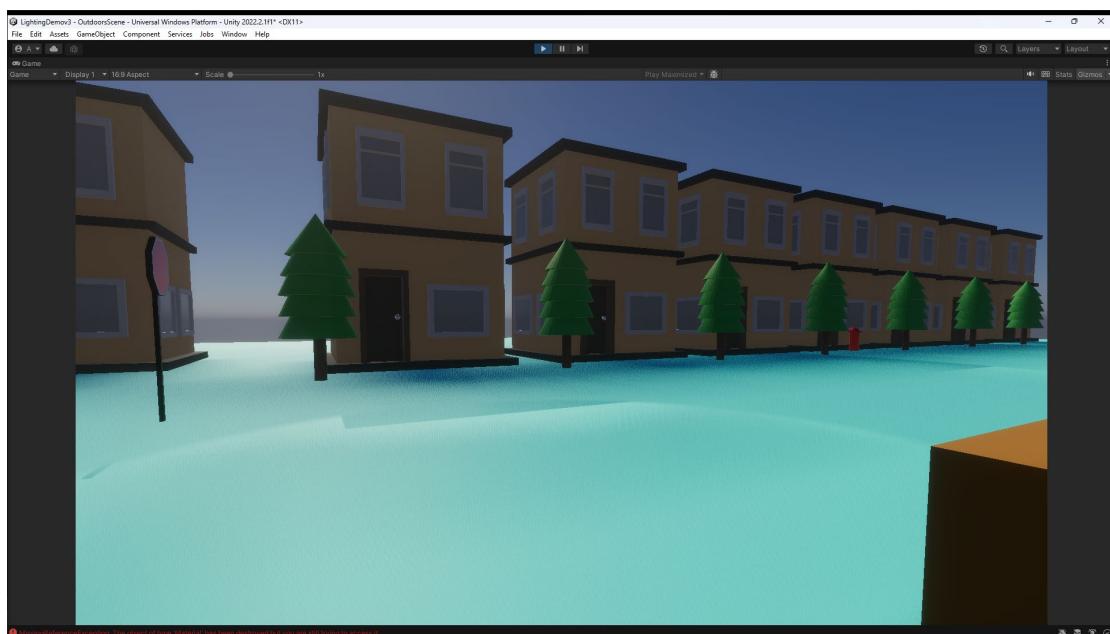
Week 3 (8)

30/01/2023 – 03/02/2023

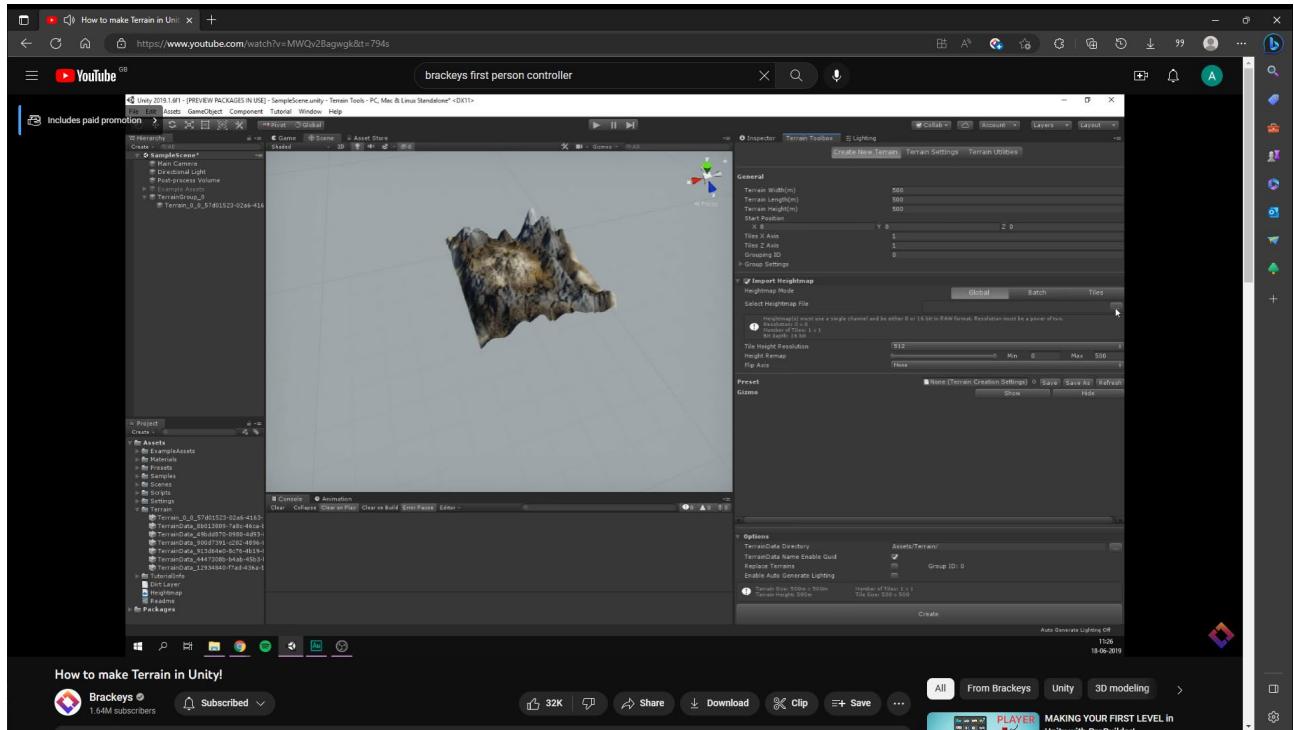
Independent Research / Study

The MVP demo was due this Friday (Demo Due Date – 03/02/2023) so I have been fully focused on this project.

I have added the models to unity and have tested that they look proper.



After the models were added, I then started to make a terrain and added vegetation to it. I followed another Brackeys tutorial to do this.



Once the terrain was, I did a test to make sure everything was working. I had a small issue when walking on the terrain, I kept falling through it. To fix this I added a box collider to it to make sure I wouldn't go through the map.

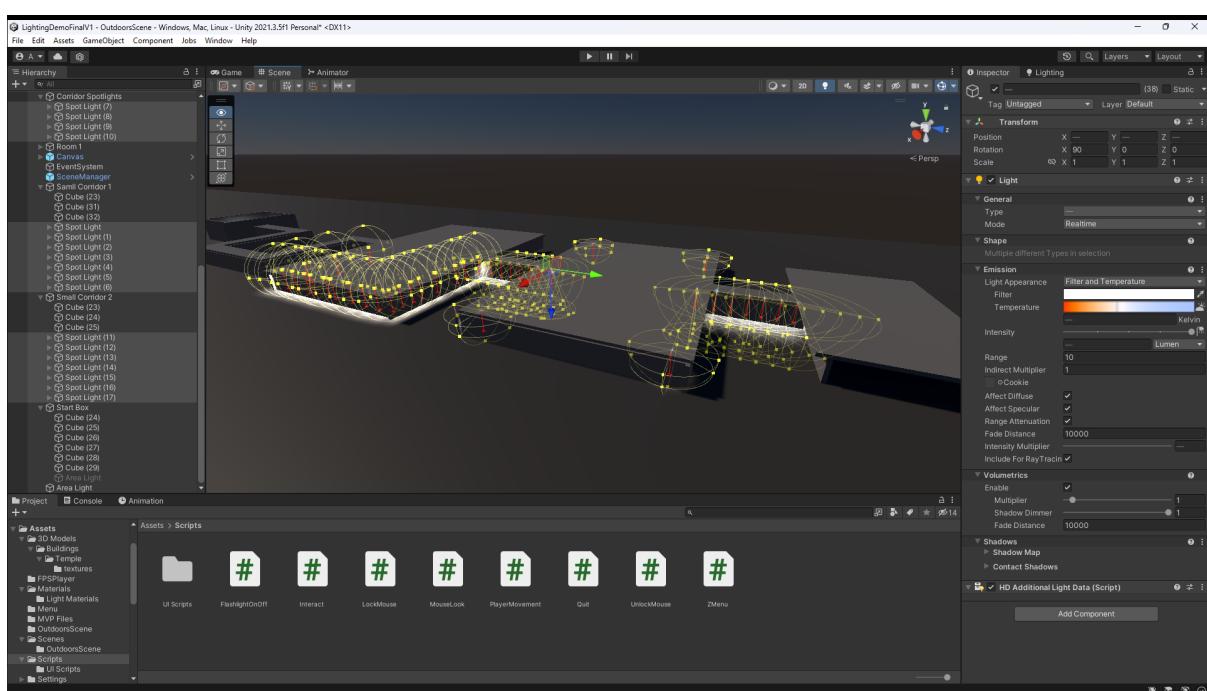
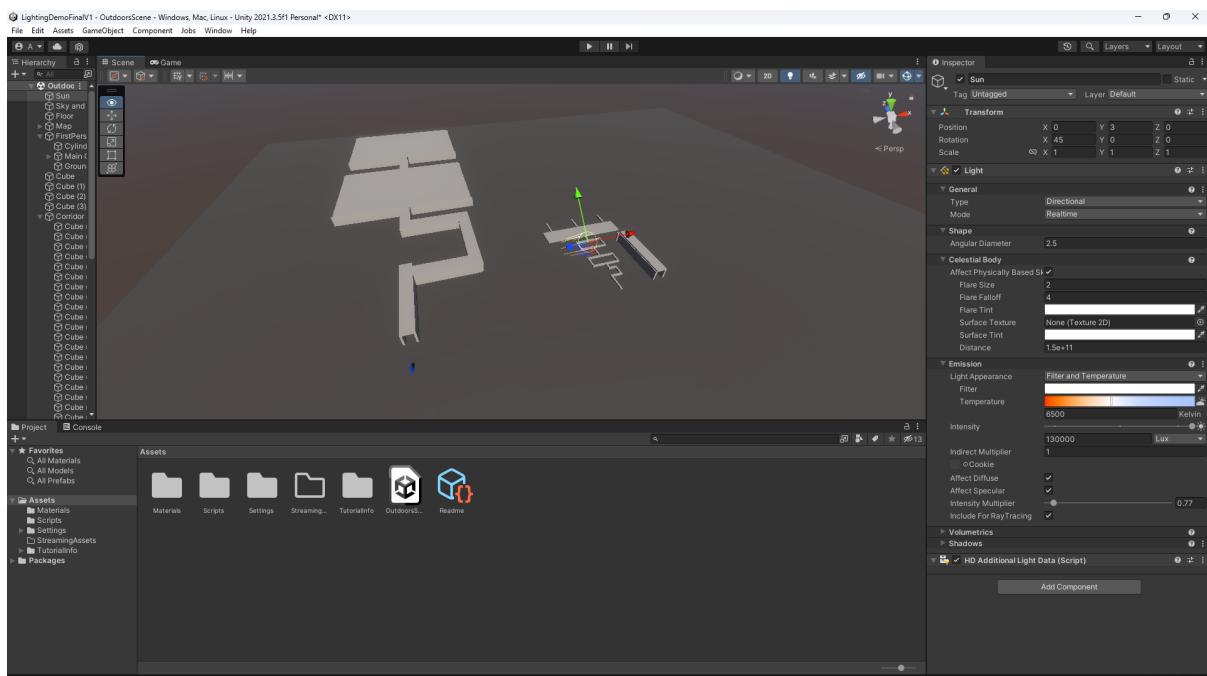
I have demonstrated the MVP to Lee, and he has stated that I am on track to finish the project. The main feedback that he wanted was that I need to think about incorporating a security aspect to the game. This will be done later down the line.

Week 4 (8)

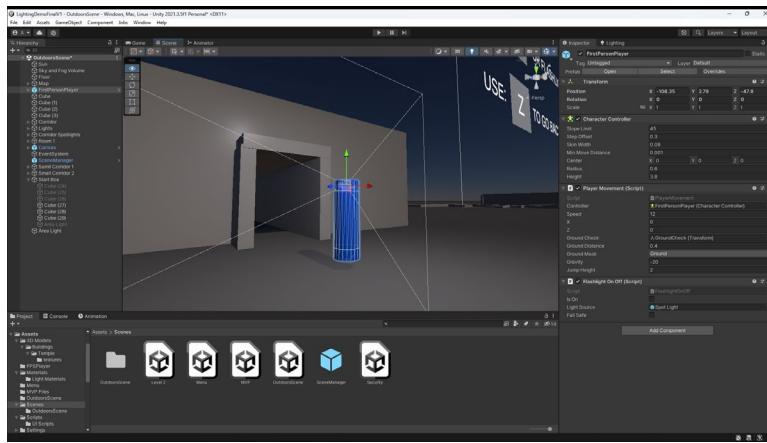
06/02/2023 – 10/02/2023

Independent Research / Study

This week I made the corridor part of level 1. I made the layout of the corridor by using cubes and expanding them out and moving them around. Then I added some area lights to the first part of the corridor. After this I added spotlights to the 2 small hallways that go from corridor 1 and 2.



After this, I added the character controller. I took the code for the MVP, but I added mouse locks to it so that it would be easier to use the UI.



After the character controller I added a flashlight to the player. There was a bug where the flash wouldn't turn off, but this was fixed by adding a delay.

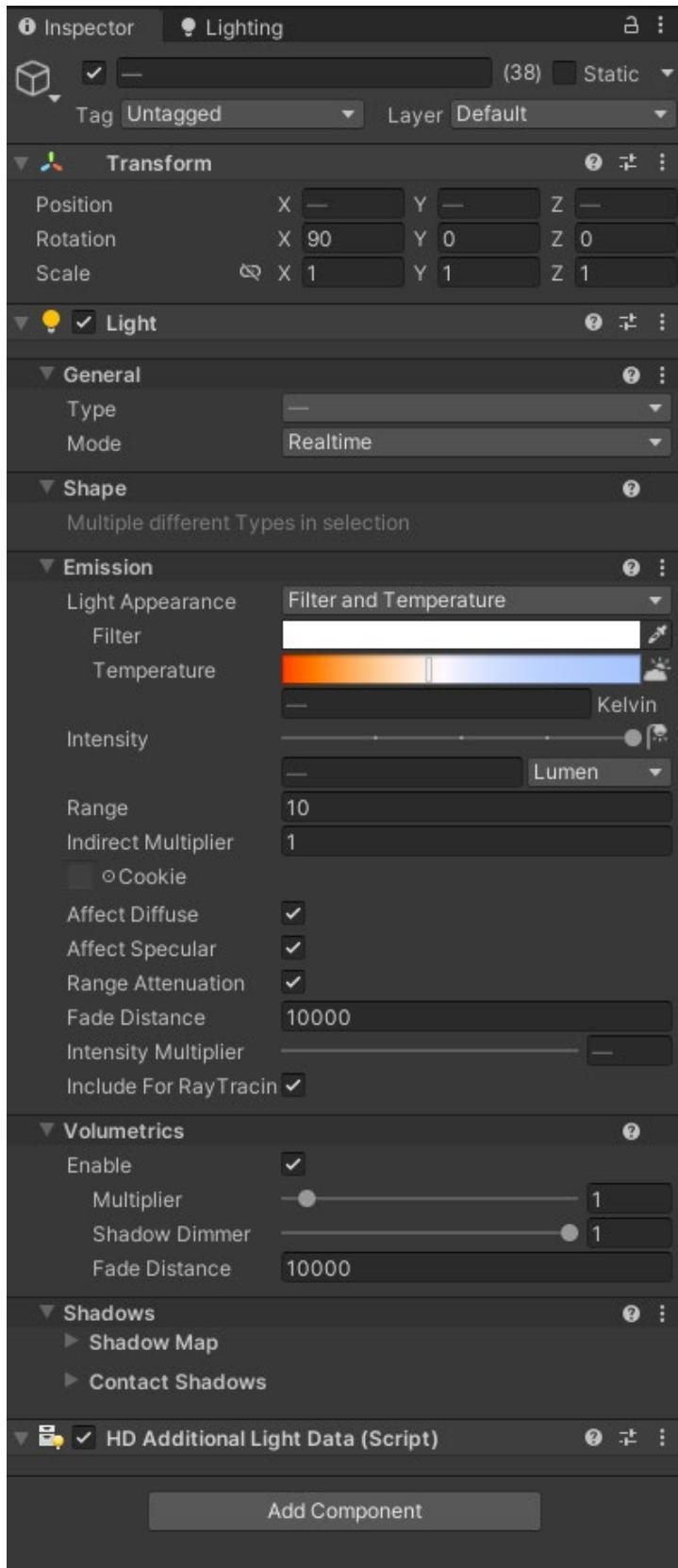
I then tested the game on my HP laptops, and they worked.

Week 5 (10)

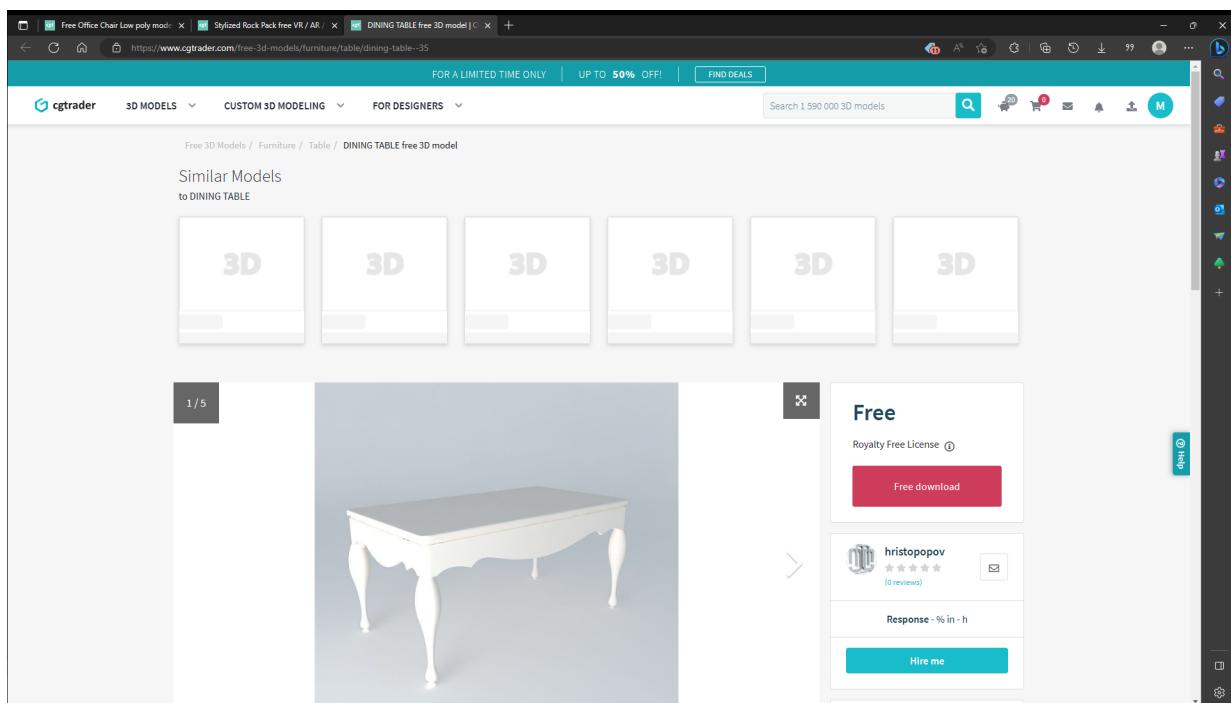
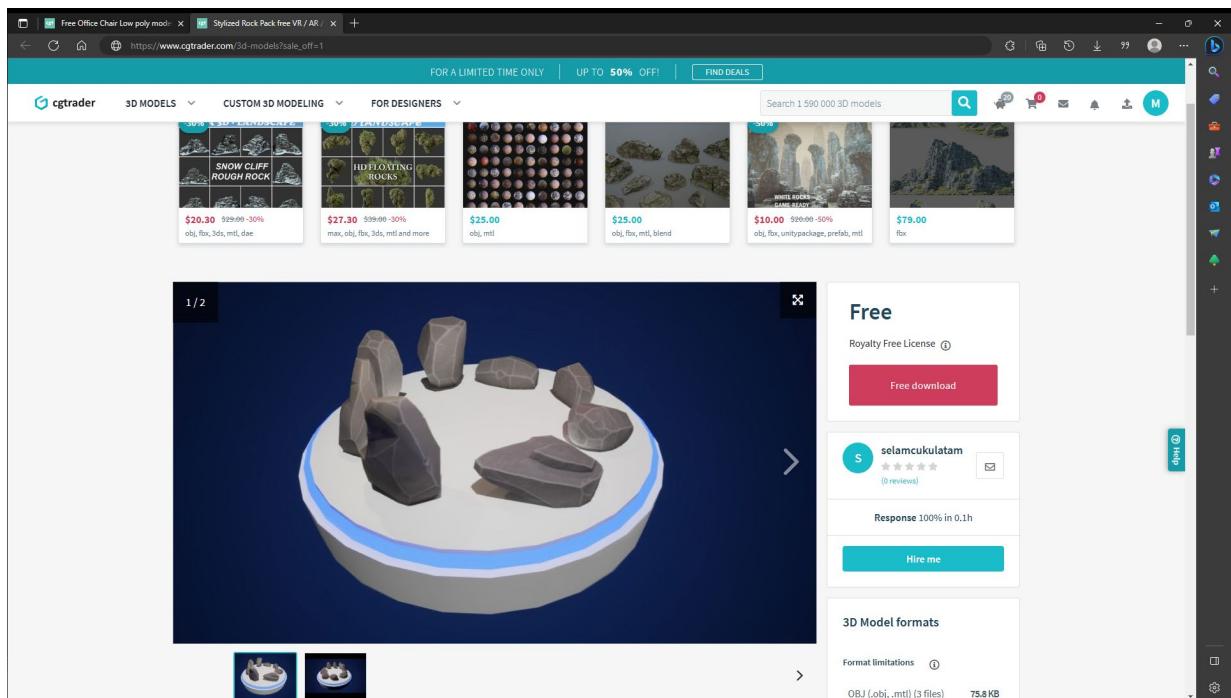
13/02/2023 – 17/02/2023

Independent Research / Study

This week I fine-tuned the lighting for the corridor by manipulating the lighting settings in unity.

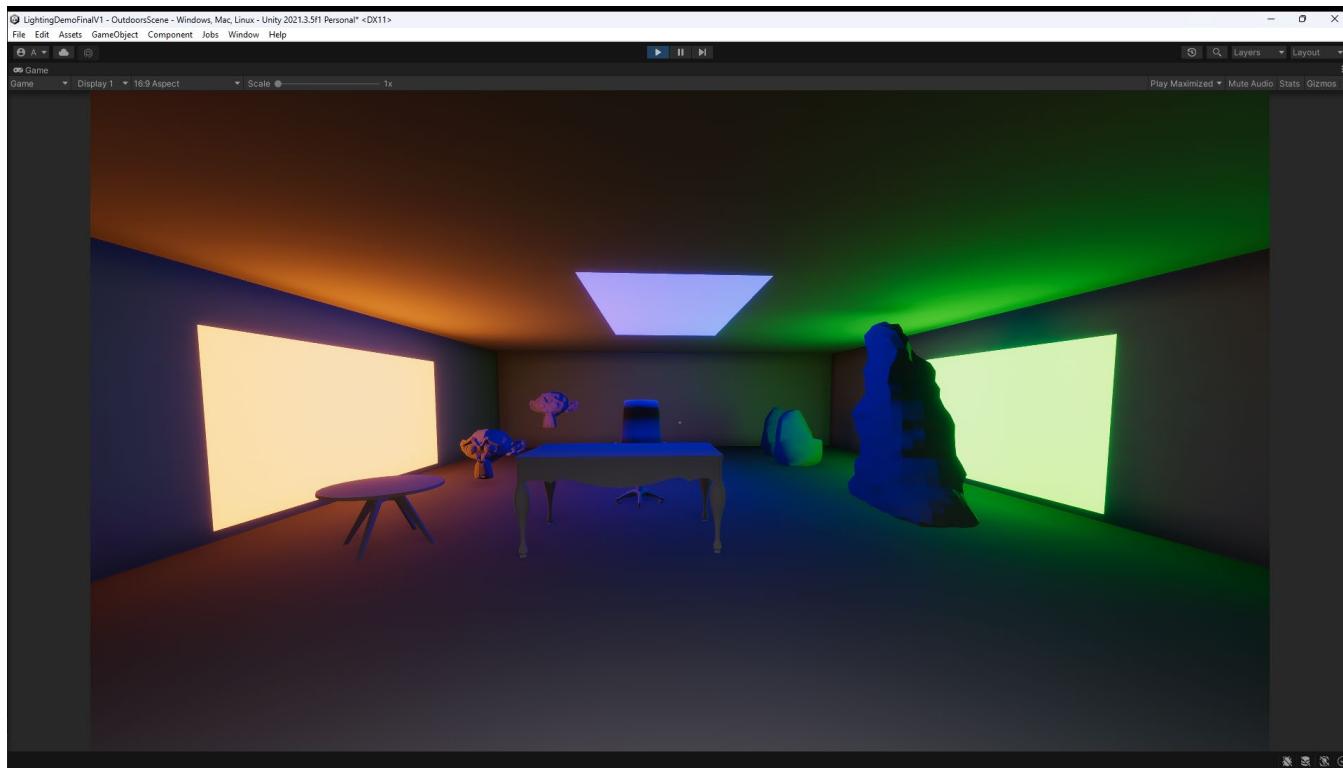


After this was completed, I started work on the lighting room which is the demonstration of baked lighting. I first downloaded 3D models from an online asset store.



Then I imported them into Unity and added them to the demo room.

Then I added pane to the left and right walls and one pane to the ceiling. Then I created an emissive material that would shine light out. After this, I attached the materials to the pane and then baked the lighting. When I first baked the lighting, it took 10 minutes as it was baking on the CPU, then I changed it to bake on the GPU. I did this a few times to get the perfect lighting and atmosphere for the room.



Week 6 (10)

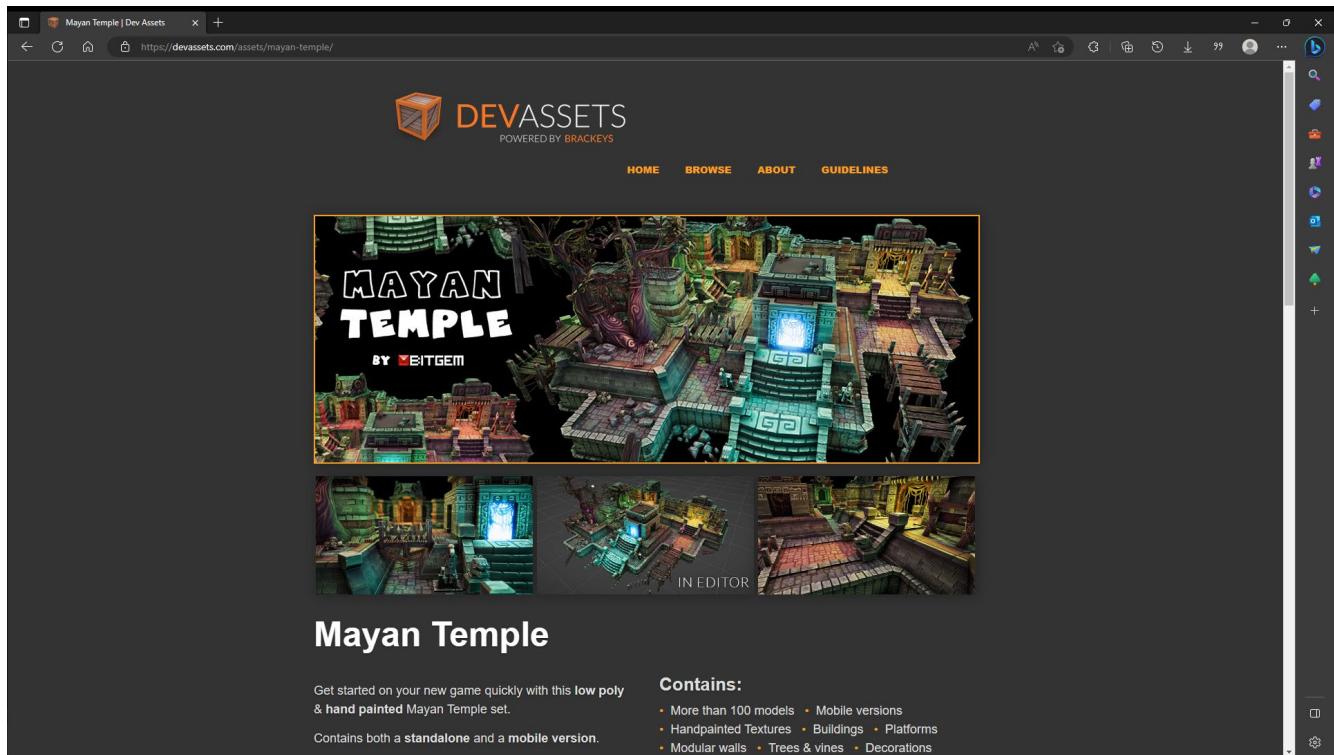
20/02/2023 – 24/02/2023

Independent Research / Study

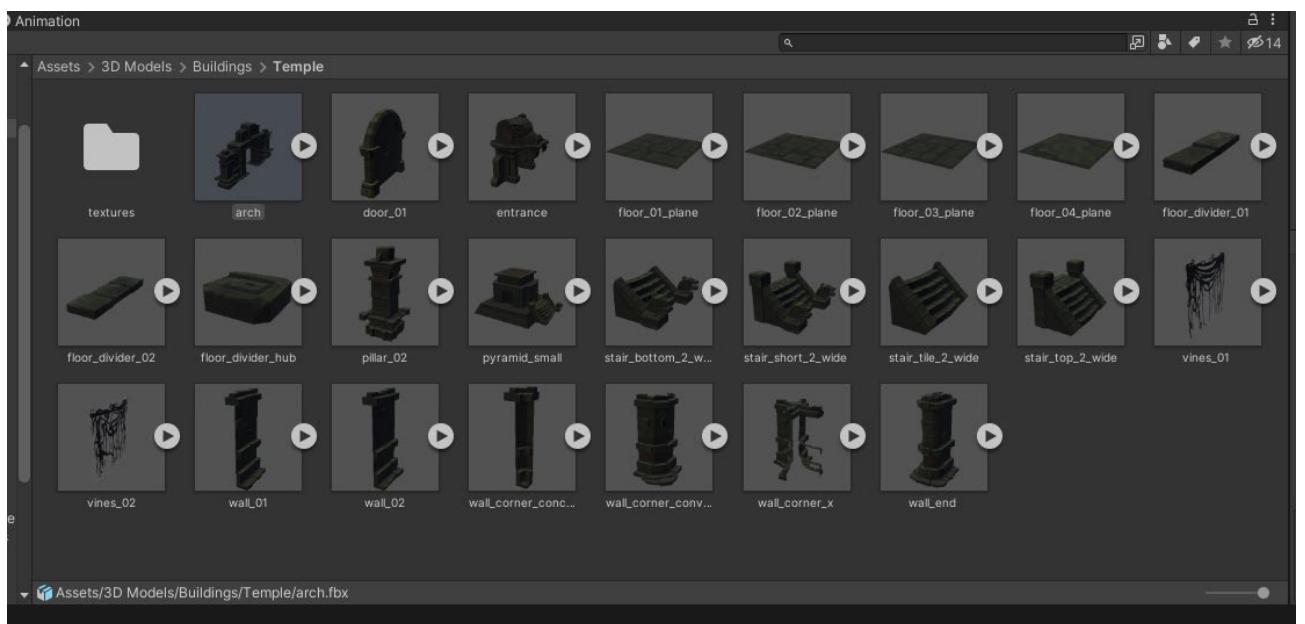
I have now started work on the second level of the demo. This level is based on real time lighting. I first added a directional light to the scene, this acts like a sun. After this I added a floor to the scene.

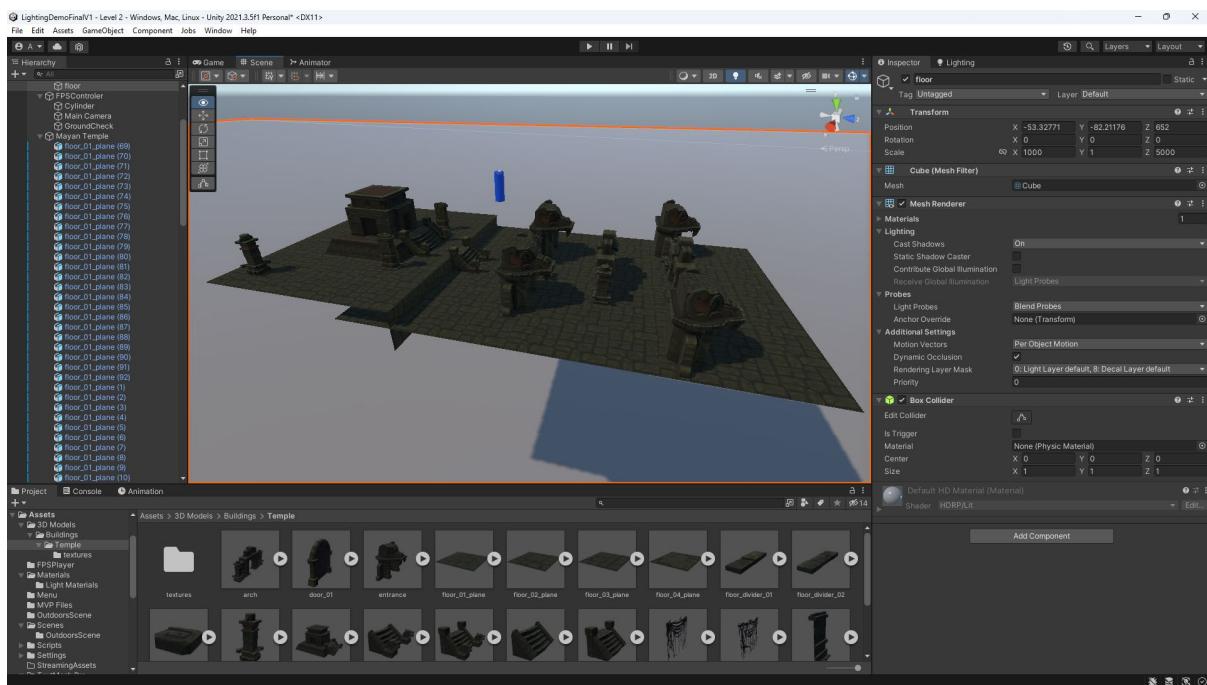
Then I created a character controller and tested the movement in the game. After this I did some research on lighting techniques in unity and how to use real time lighting.

After researching, I needed to create a map for the player to walk through. This was done by using premade assets from a website called 'DevAssets'.



The downloaded model gave lots of small to make a temple in any way that I like.





This is the map that I made using the premade assets. This took quite some time and lots of trial and error. After this was made, I tested it on my HP laptops to see if they worked which it has worked.

Week 7 (8)

27/02/2023 – 03/03/2023

Dr Julian Bass gave a lecture on dissertation writing.

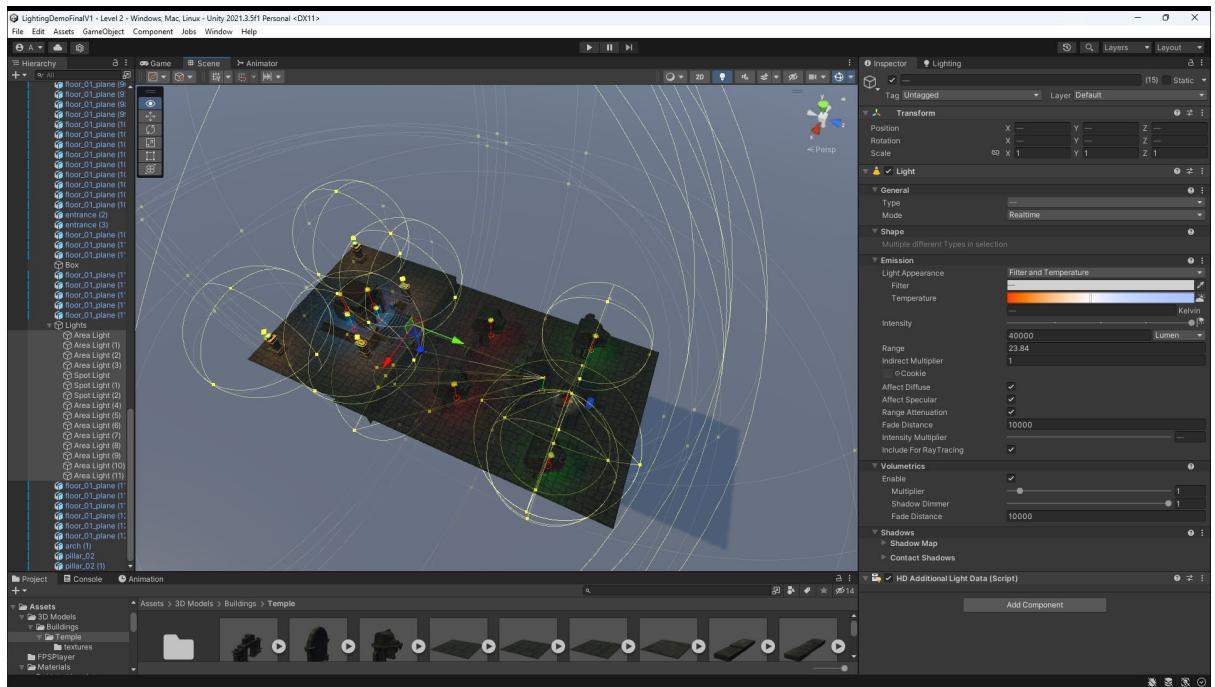
Julian talked about how we should structure our dissertation. He also requests that we keep making drafts and have the drafts checked with our supervisor. He also showed a template structure of a dissertation. So far, I have followed the template for this. He also states that we should contact our supervisors about our heading and sub-headings to get the correct ones.

Julian recommends that we apply a lean software methodology to our writing, which is, eliminate waste, aim for perfection, features are pulled, flow. He also gave 2 writing processes, which are top down and bottom up.

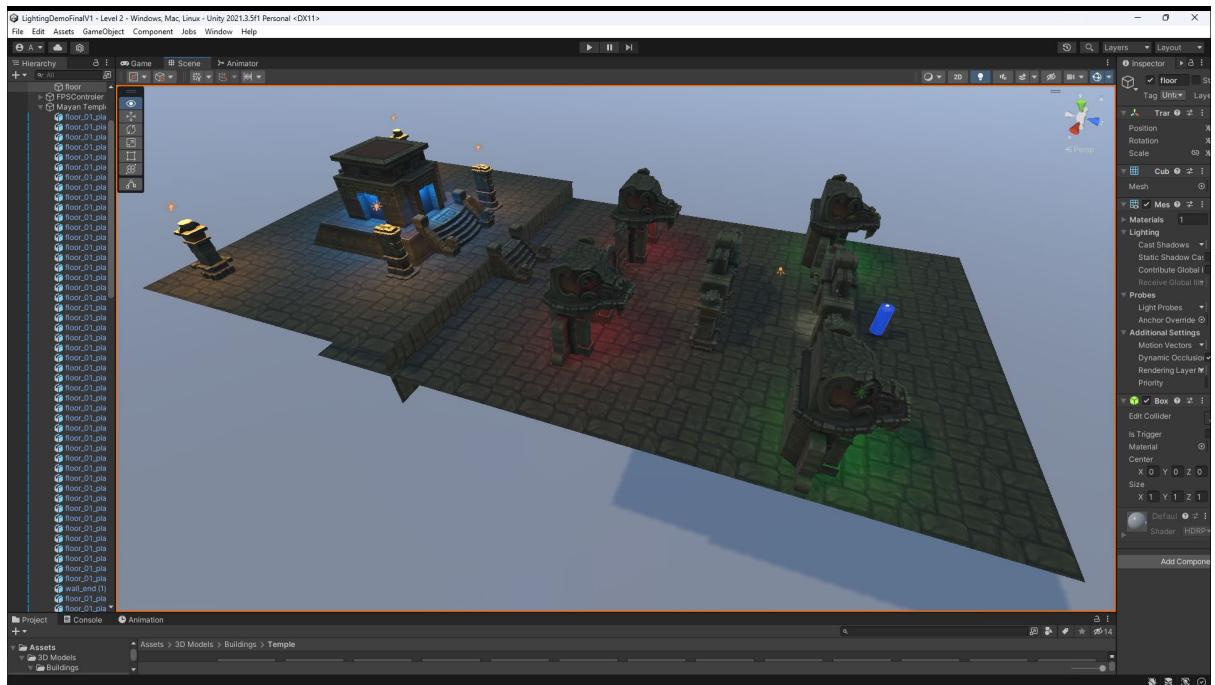
Julian also talks about science writing, which is evidence based, factual, carefully organized and has a logical flow of arguments. Julian also talks about the writing style of a dissertation, which includes, make paragraph unit of composition, begin each paragraph with a topic sentence. Use active voice, remove needless words, write in the positive and be concrete.

Independent Research / Study

This week I added all the lighting to the scene and all the box colliders to the scene so the player can walk in the temple.



These are the area lights and spotlights with the different color lighting in the level.



This is the final look of level 2 after it has been completed.

Week 8 (8)

06/03/2023 – 10/03/2023

Independent Research / Study

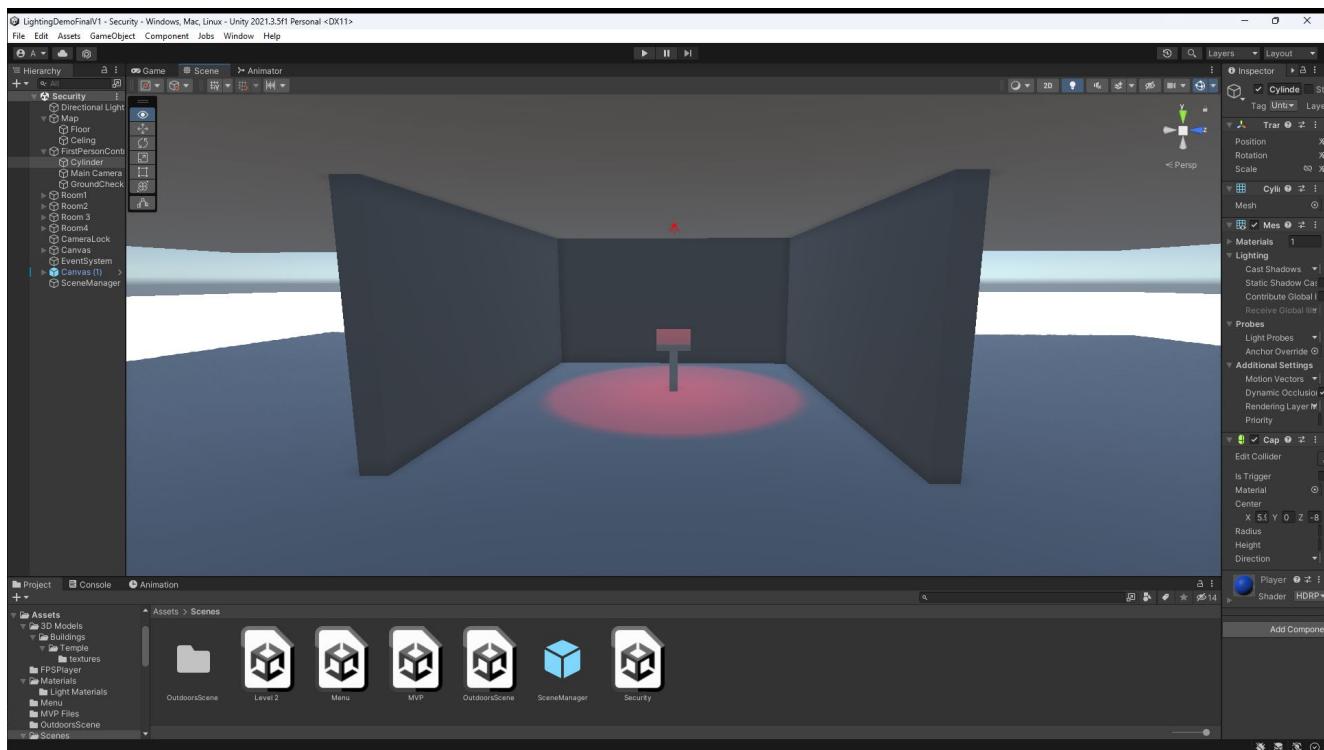
Level 3 and dissertation

This week I have divided my attention into 2 parts, the first half I focused on getting level 3 done, the second half I focused on the design and requirements part of the dissertation.

This level is quite a simple level in graphics and lighting, as this level focuses on the security aspects of the project. Their main idea for this level is to inform the player about the security issue and how to avoid them. To do this the player needs to walk up to an altar and interact with it and it will bring up a dossier of information.

The first thing that I did was do research on security new stories. I picked 4 of the stories, GTA 5 Online hacks, CDProjekt Red source code Leak, Call of Duty: Black Ops 3 remote access trojan hacks and Sony PlayStation live hack.

Then I created the altars where these stories should go.



The second half was focused on the dissertation and trying to get the design and requirements part finished.

Week 9 (10 Hours)

13/03/2023 – 17/03/2023

Dr Norman Murray gave a lecture on Final Year Projects Deliverables

Norman has given a layout of the dissertation which is:

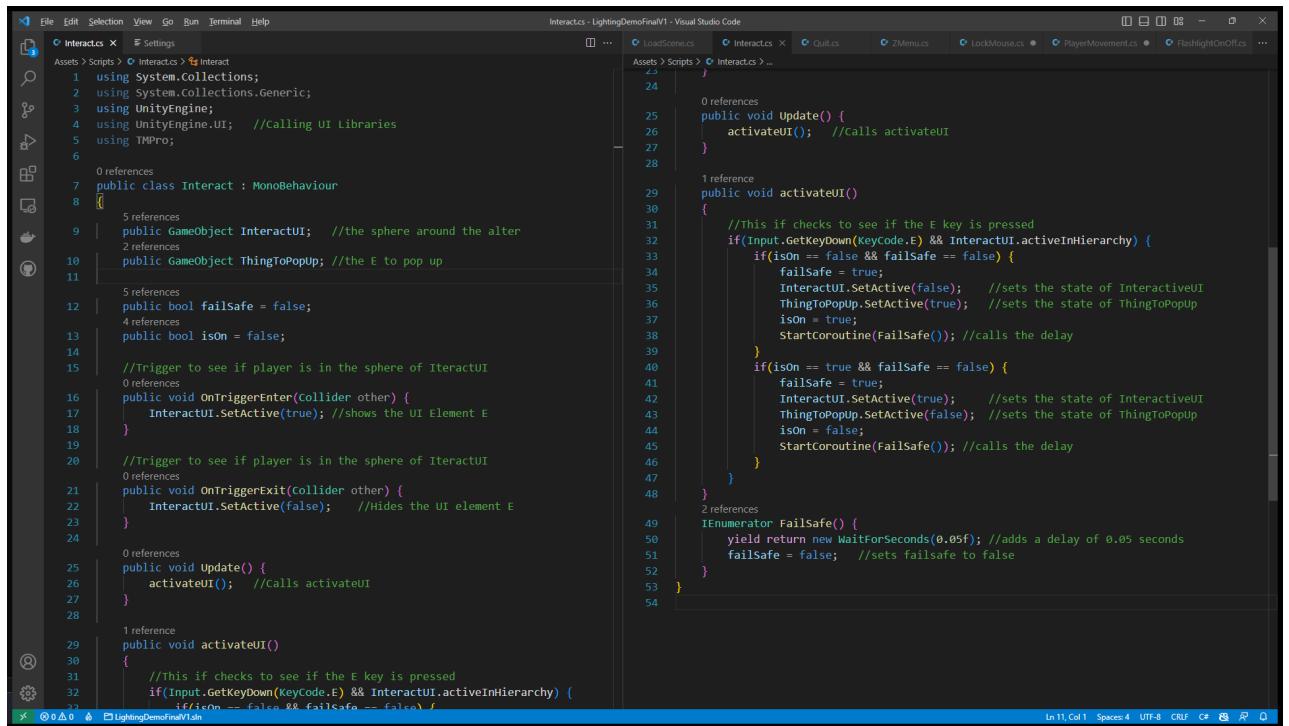
- Abstract
- Introduction
- (Literature Review)
- (Methodology)
- Requirements
- Design
- Implementation
- Testing
- Critical Evaluation
- Conclusions

Norman also explains each section of the dissertation. He also gave a timetable for the demonstration which is going to be scheduled for week 12 of trimester 2, 24th April 25th April. 26th April is a backup date. He talks about the 3 parts of the demo which are, 5 -10 minutes presentation, a 5 - 10 minutes demonstration on what has been developed and 5 minutes for questions.

Independent Research / Study

Near the end of the week, I began to feel ill, and then I found out that I had gotten covid. Sick from 16/03/2023 – 22/03/2023

This week I finished level 3 by adding the stories to the game and implementing an interactable script to the scene, so when the player walks near the altar, a UI element will pop up that says press the 'E' key.



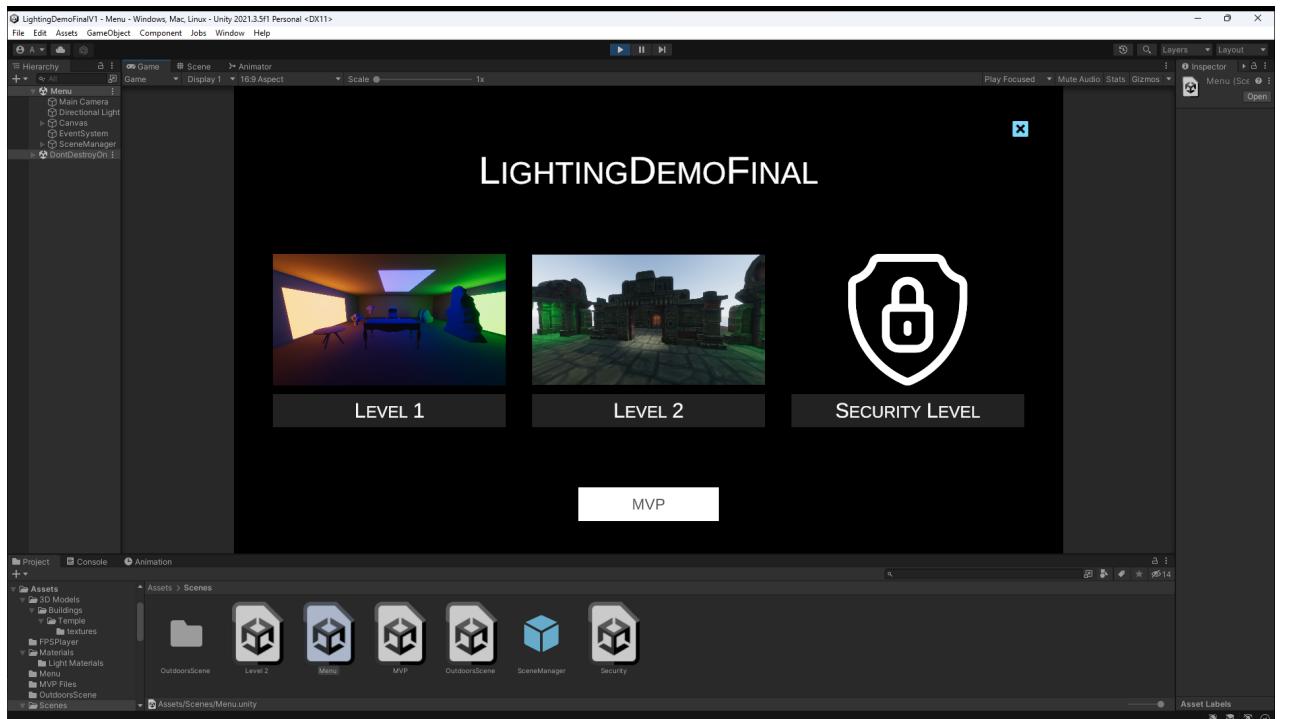
```

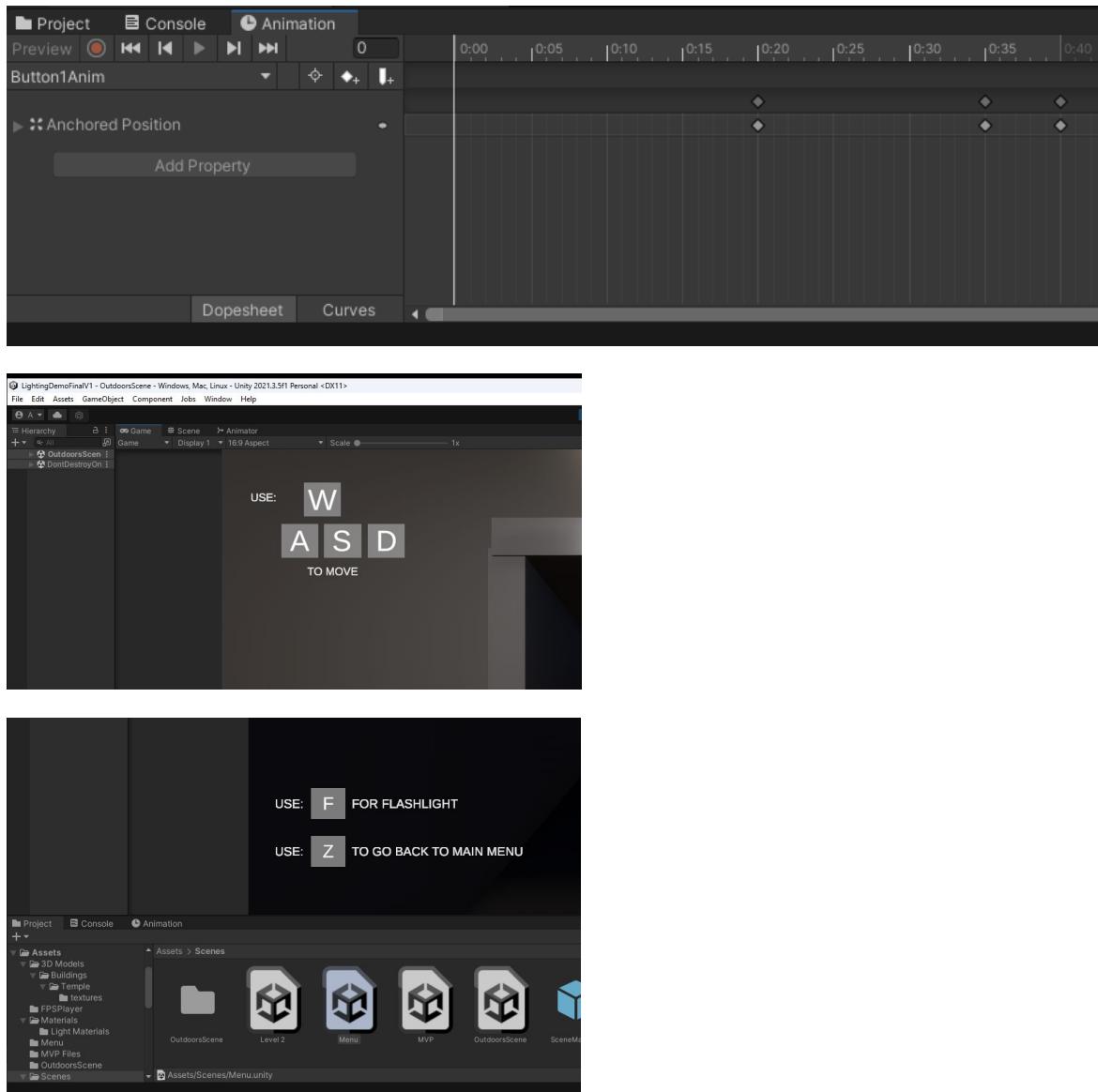
Interact.cs - LightingDemoFinalV1 - Visual Studio Code

Assets > Scripts > Interact > Interact
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4 using UnityEngine.UI; //Calling UI Libraries
5 using TMPro;
6
7 public class Interact : MonoBehaviour
8 {
9     public GameObject InteractUI; //the sphere around the alter
10    public GameObject ThingToPopUp; //the E to pop up
11
12    public bool failsafe = false;
13    public bool isOn = false;
14
15    //Trigger to see if player is in the sphere of InteractUI
16    public void OnTriggerEnter(Collider other) {
17        InteractUI.SetActive(true); //shows the UI Element E
18    }
19
20    //Trigger to see if player is in the sphere of InteractUI
21    public void OnTriggerExit(Collider other) {
22        InteractUI.SetActive(false); //Hides the UI element E
23    }
24
25    public void Update() {
26        activateUI(); //Calls activateUI
27    }
28
29    public void activateUI()
30    {
31        //This if checks to see if the E key is pressed
32        if(Input.GetKeyDown(KeyCode.E) && InteractUI.activeInHierarchy) {
33            isOn = false && failsafe == false) {
34
35            failsafe = true;
36            InteractUI.SetActive(false); //sets the state of InteractUI
37            ThingToPopUp.SetActive(true); //sets the state of ThingToPopUp
38            isOn = true;
39            StartCoroutine(failsafe()); //calls the delay
40
41        if(isOn == true && failsafe == false) {
42            failsafe = true;
43            InteractUI.SetActive(true); //sets the state of InteractUI
44            ThingToPopUp.SetActive(false); //sets the state of ThingToPopUp
45            isOn = false;
46            StartCoroutine(failsafe()); //calls the delay
47
48        }
49    }
50    IEnumerator failsafe()
51    {
52        yield return new WaitForSeconds(0.05f); //adds a delay of 0.05 seconds
53        failsafe = false; //sets failsafe to false
54    }
}

```

After this I added a main menu to the game and then I added the animation for the main menu as well. Then I add a HUD to all the levels where it gives instructions on how to walk and turn on the flashlight. There is an animation on this as well, where the instruction goes off the screen after a few seconds.





I also finished the design and requirements part of the dissertation.

Week 10 (3)

20/03/2023 – 24/03/2023

Independent Research / Study

Sick from 15/03/2023 – 21/03/2023

After recovering from Covid, I checked my dissertation and then I emailed it to Lee.

I then carried on working on the other parts of the dissertation.

Easter Holidays - 27/03/2023 – 16/04/2023 (32 hours)

I went to university on the 28/03/2023 to do a test the lighting demo on the University computers, unfortunately the demo did not run on the university computers as the unity version where mismatched and the built version of the game didn't run because the permissions didn't work. This is when I became really stressed as I didn't have a powerful enough portable machine to take with me to the demonstration in late April. I decided to go back home and port my entire demo over to the University installed version of unity which is version 2021.3.5f1.

After porting the game, I went back to university on 04/04/2023 to test the lighting demo again. Luckily this time it had worked, and the game ran successfully.



After testing the game in university, I finished off my dissertation.

Emailed the full dissertation to lee for feedback.

Week 11

17/04/2023 – 21/04/2023

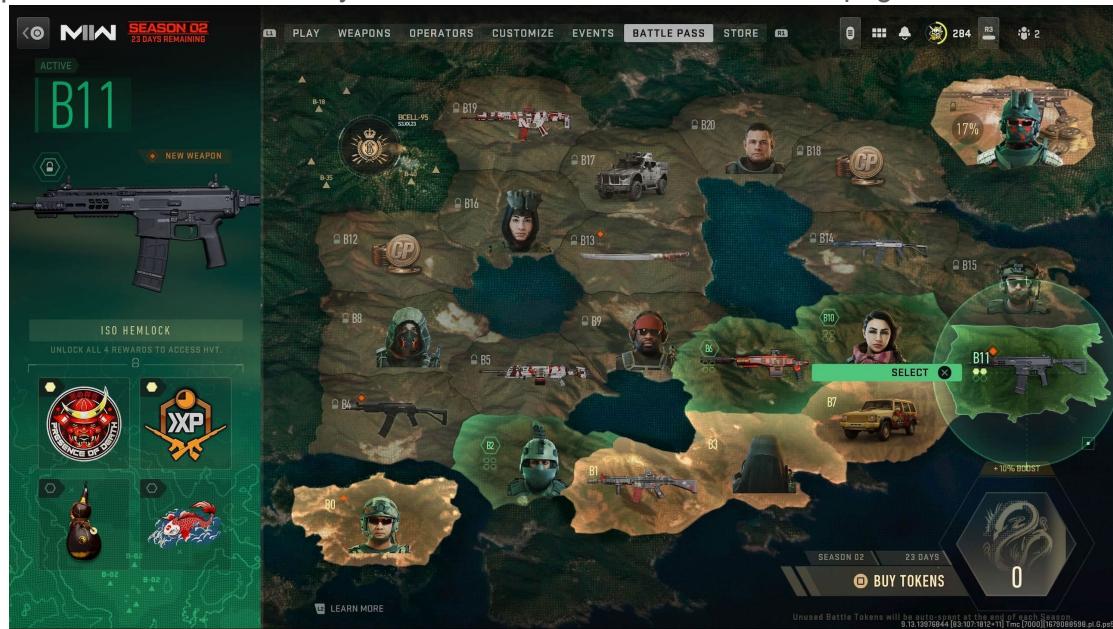
17/04/2023 is dissertation and project submission date.

Week 12 is Demonstration Monday April 24th and Tuesday April 25th.

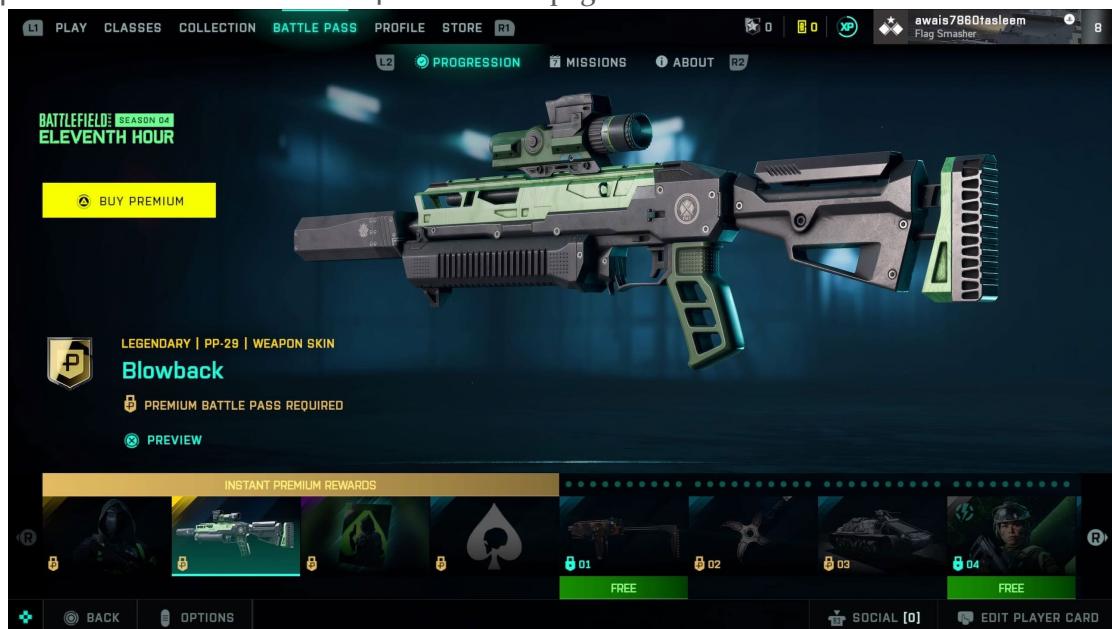
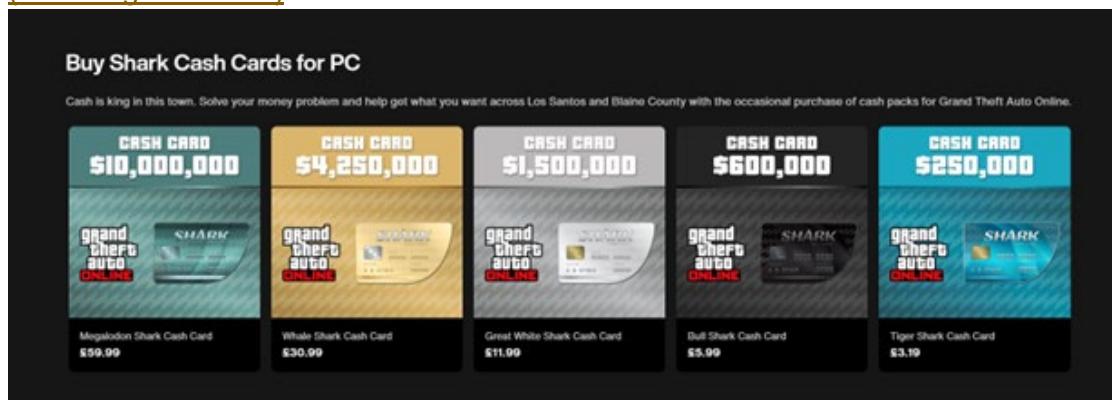
Appendix c: PlayStation 5 Box



Appendix d: Call of Duty: Modern Warfare 2 2022 Battle Pass page:



Appendix e: Battlefield 2042 Battle Pass page:

Appendix f: Shark Cash Card prices and link to the website - rockstargames.com

Appendix g: GTA 5 Mod Menu

UnKnoWnCheaTs.com - Multiplayer Game Hacking and Cheats - Spectrum FREE

GTA V Menu



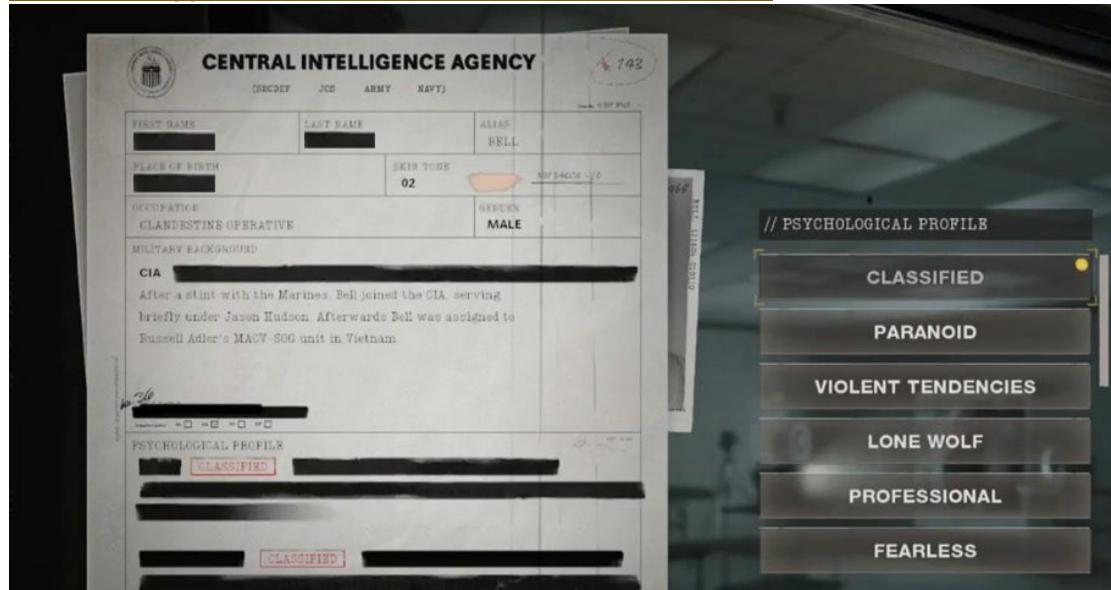
Appendix h: Minecraft Signpost

https://www.seekpng.com/png/u2w7e6i1u2i1a9t4_overview-signedit-mods-projects-minecraft-sign-png/



Appendix i: Black Ops: Cold War Dossier

<https://www.qgrecon.com/articles/cold-war-character-trait/>



Appendix j: [Figma Downloads | Web Design App for Desktops & Mobile](#)

Appendix k: Draw.IO Link [diagrams.net](https://www.diagrams.net)

Appendix I: Jira Support chat 1

The image shows a Jira support ticket and a Jira support chat. The ticket is for issue IN-28388, titled "Crash on launch everytime". The ticket details include:

- Description:** 1. What happened: Opening my project crashes unity, its a fresh new hdrp project with a cube in the center. No scripts and other stuff. I have resatred my windows pc a few times and the error still comes up.
2. How can we reproduce it using the example you attached: open the project from unity hub and it will crash when trying to open the scene after it has done the compiling process and the unity UI shows up.
- Correspondent:** awaist7860@hotmail.com
- CPU:** Intel(R) Core(TM) i5-8400 CPU @ 2.80GHz
- GPU:** NVIDIA GeForce GTX 1060 6GB NVIDIA GeForce GTX 1060 6GB NVIDIA GeForce GTX 1060 6GB
- Operating System:** Windows 11 (10.0.22621) 64bit
- Selected Platform:** WindowsStandaloneSupport
- Reviewer:** Daugardas Lukšas

The ticket status is **CLOSED**. Notifications are off. The request type is **Incoming Incident**. The ticket is shared with awaist7860@hotmail.com and Unity3d. There are links to various logs and files.

The Jira support chat shows the following messages:

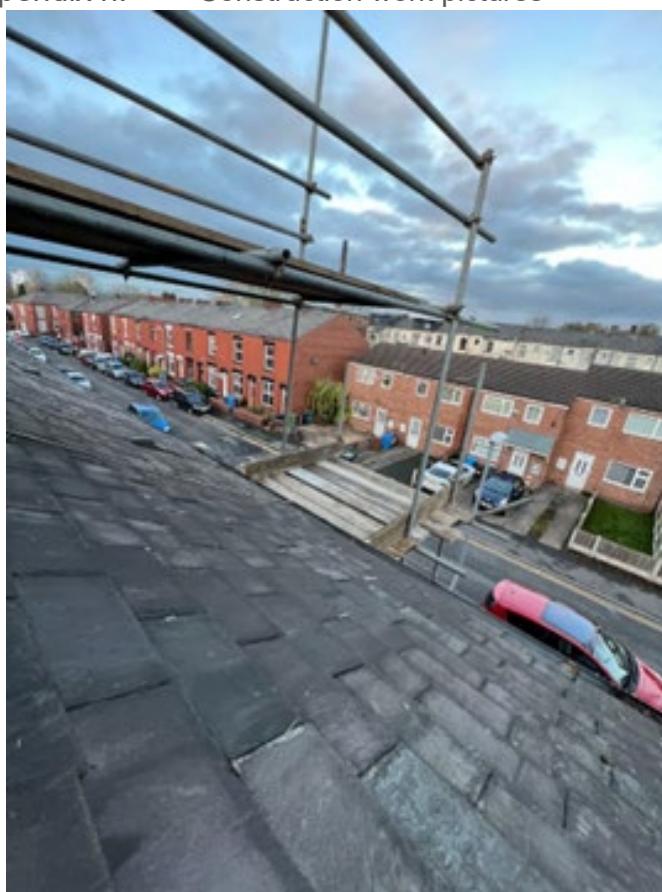
- Automatic response 10/Jan/23 1:14 PM: Your request status has changed to In Review.
- DL Daugardas Lukšas 10/Jan/23 1:15 PM: Hi,
Thanks for getting in touch, we actually know about this issue and you can check the status here:
<https://issuetracker.unity3d.com/product/unity/issues/guid/UUM-12979>
This issue is currently fixed in 2023.1.0a25 and above.
Please reach out to me if I can answer any questions or be of further help.
Thanks,
Daugardas
Customer QA Team
- Automatic response 10/Jan/23 1:15 PM: Your request status has changed to Closed with resolution Duplicate.

Add a comment

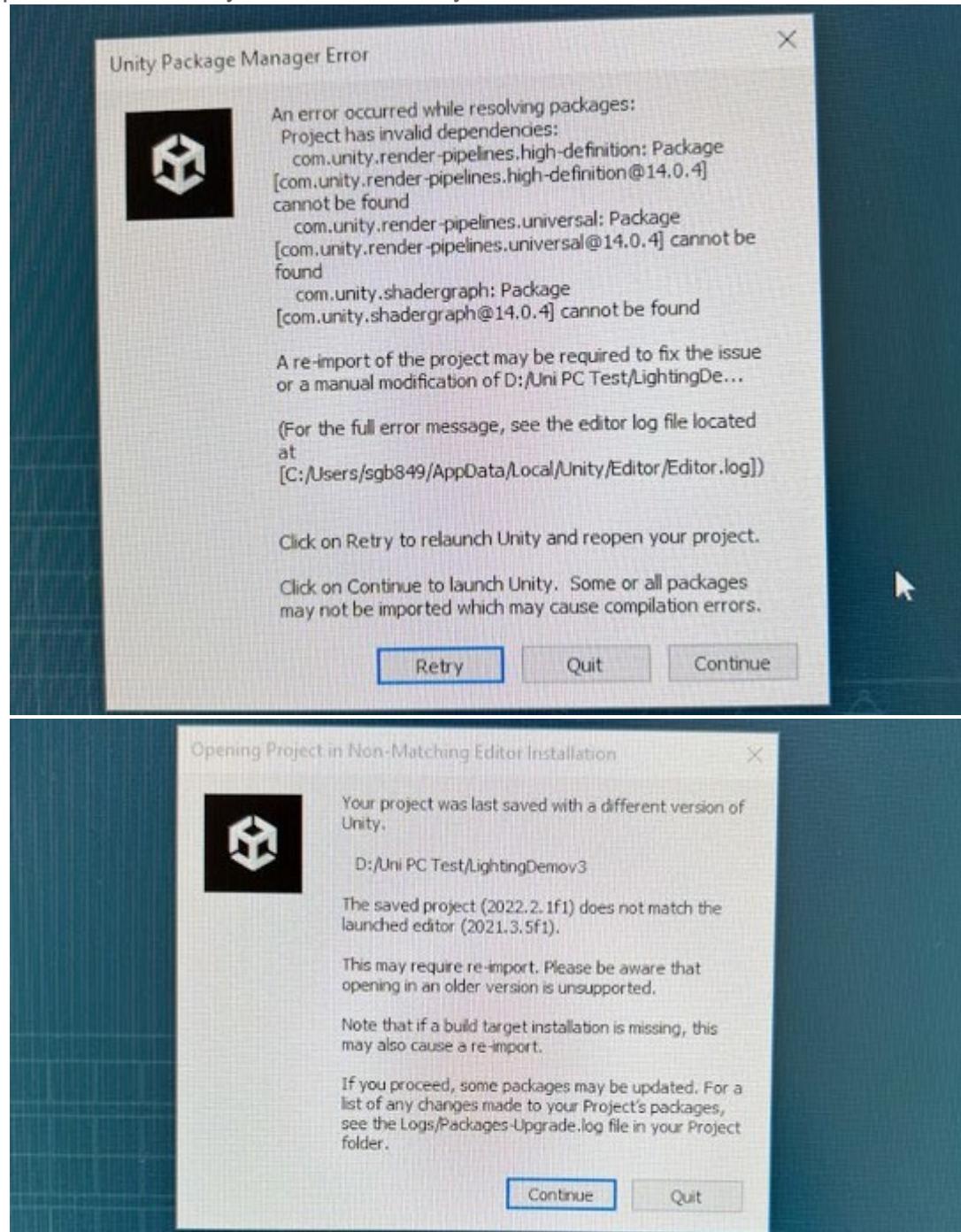
Appendix m: New AMD Ryzen CPU

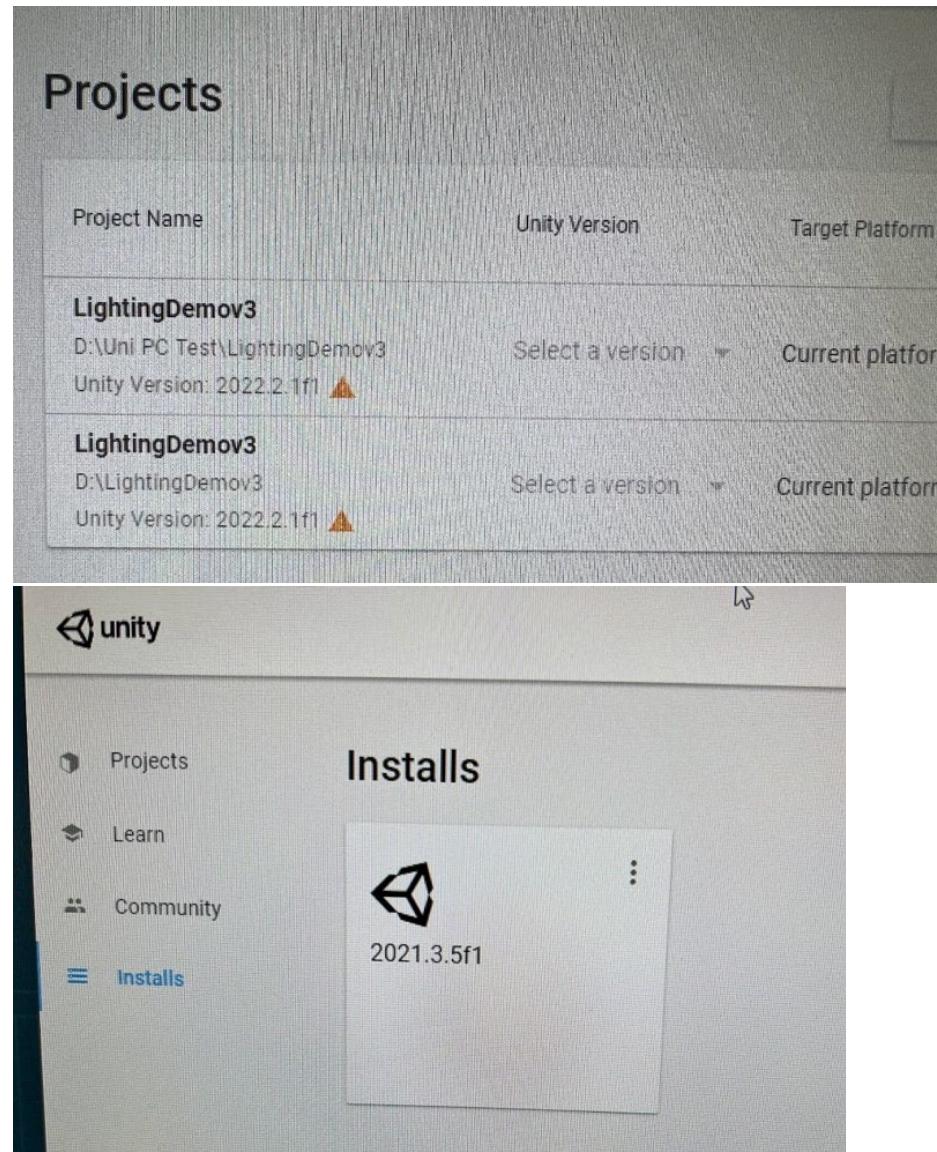


Appendix n: Construction work pictures



Appendix o: Unity Errors in University





The image shows the Unity Hub application interface. At the top, there is a table with two rows of project information. The columns are labeled "Project Name", "Unity Version", and "Target Platform".

Project Name	Unity Version	Target Platform
LightingDemov3 D:\Uni PC Test\LightingDemov3 Unity Version: 2022.2.1f1 ⚠	Select a version	Current platform
LightingDemov3 D:\LightingDemov3 Unity Version: 2022.2.1f1 ⚠	Select a version	Current platform

Below the table, the Unity Hub navigation bar is visible, showing "unity" and the "Installs" section. The "Installs" section displays a single installed version of Unity: "2021.3.5f1".